UNCLASSIFIED

AD NUMBER
ADB102656
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to DoD and DoD contractors only; Critical Technology; SEP 1985. Other requests shall be referred to Air Force Armament Lab., Eglin AFB, FL 32542. This document contains export-controlled technical data.**
AUTHORITY
AFSC/mnoi's wright lab, eglin afb,fl; ltr dtd 13 Feb 1992

THIS PAGE IS UNCLASSIFIED

AFATL-TR-85-93

Common Ada® Missile Packages (CAMP)

Volume III: Part Rationales

Daniel G. McNicholl
Constance Palmer, et al.

McDONNELL DOUGLAS ASTRONAUTICS COMPANY
POST OFFICE BOX 516
ST. LOUIS, MISSOURI 63166

MAY 1986

FINAL REPORT FOR PERIOD SEPTEMBER 1984 - SEPTEMBER 1985

AIR FORCE ARMAMENT LABORATORY
Air Force Systems Command • United States Air Force • Eglin Air Force Base, Florida

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any potential invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Donald C Daniel

DONALD C. DANIEL
Chief, Aeromechanics Division

Even though this report may contain special release rights held by the controlling office, please do not request copies from the Air Force Armament Laboratory. If you qualify as a recipient, release approval will be obtained from the originating activity by DTIC. Address your request for additional copies to:

Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22314

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization, please notify AFATL/FXG, Eglin AFB, FL 32542

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Distribution limited to DOD and DOD Contractors only; this report documents ~~test and evaluation~~ distribution limitation applied September 1985. C/T
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S): AFATL-TR- 85-93
6a NAME OF PERFORMING ORGANIZATION: McDonnell Douglas Astronautics Company; 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION: Aeromechanics Division Air Force Armament Laboratory
6c ADDRESS (City, State and ZIP Code): P. O. Box 516 St. Louis, MO 63166	7b ADDRESS (City, State and ZIP Code): Eglin AFB, FL 32542-5434
8a NAME OF FUNDING/SPONSORING ORGANIZATION: STARS Joint Program Office; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F08635-84-C-0280
8c ADDRESS (City, State and ZIP Code): Room 3D139 (1211 Fern St.) The Pentagon Washington, D.C. 20301-3081	10 SOURCE OF FUNDING NOS: PROGRAM ELEMENT NO. 63756A; PROJECT NO.; TASK NO.; WORK UNIT NO.
11 TITLE (Include Security Classification): Common Ada® Missile Packages (CAMP), Volume III	

12 PERSONAL AUTHOR(S): McNicholl, Daniel G., Palmer, Constance, Cohen, Sanford G., Whitford, William H., Goeke, Gerard O.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Yr., Mo., Day)	15 PAGE COUNT
FINAL	FROM Sep 84 TO Sep 85	May 1986	95

16 SUPPLEMENTARY NOTATION: SUBJECT TO EXPORT CONTROL LAWS. Availability of this report is specified on verso of front cover.

17 COSATI CODES		
FIELD	GROUP	SUB. GR.

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Reusable Software, Missile Software, Software Generators, Ada, Parts Composition, Systems, Software Parts.

19 ABSTRACT (Continue on reverse if necessary and identify by block number) The objective of the CAMP program is to demonstrate the feasibility of reusable Ada software parts in a real-time embedded application area; the domain chosen for the demonstration was that of missile flight software systems. This required that the existence of commonality within that domain be verified (in order to justify the development of parts for that domain), and that software parts be designed which address those areas identified. An associated parts cataloging scheme and parts composition system were developed to support parts usage.

see p1

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS ☒	21 ABSTRACT SECURITY CLASSIFICATION: Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL: Christine Anderson	22b TELEPHONE NUMBER (Include Area Code): (904) 882-2961; 22c OFFICE SYMBOL: AFATL/ FXG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

11. TITLE (CONCLUDED)

Part Rationales

3. DISTRIBUTION/AVAILABILITY OF REPORT. (CONCLUDED)

Other requests shall be referred to the Air Force Armament Laboratory (FXG), Eglin Air Force Base, Florida 32542-5434.

PREFACE

This report describes the work performed, the results obtained, and the conclusions reached during the Common Ada Missile Packages (CAMP) contract (F08635-84-C-0280). This work was performed by the Computer Systems & Software Engineering Department of the McDonnell Douglas Astronautics Company, St. Louis, Missouri (MDAC-STL) and was sponsored by the United States Air Force Armament Laboratory (FXG), at Eglin Air Force Base, Florida. This contract was performed between September 1984 and September 1985.

The MDAC-STL CAMP program manager was Dr. Daniel G. McNicholl (McDonnell Douglas Astronautics Company, Computer Systems and Software Engineering Department, P.O. Box 516, St. Louis, Mo. 63166) and the AFATL CAMP program manager was Christine M. Anderson (Air Force Armament Laboratory, Aeromechanics Division, Guidance and Control Branch, Eglin Air Force Base, Florida 32542).

This report consists of three volumes. Volume I contains overview material and the results of the CAMP commonality study. Volume II contains the results from the CAMP automated parts engineering study. Volume III contains the rationale for the CAMP parts.

Commercial hardware and software products mentioned in this report are sometimes identified by manufacturer or brand name. Such mention is necessary for an understanding of the R & D effort, but does not constitute endorsement of these items by the U.S. Government

ACKNOWLEDGEMENT

Special Thanks to the Armament Division Standardization Office and to the Software Technology for Adaptable, Reliable Systems (STARS) Joint Program Office for their support of this project.

(The reverse of this page is blank)

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONCLUDED)

(The reverse of this page is blank)

LIST OF FIGURES

LIST OF FIGURES (Concluded)

SECTION I
INTRODUCTION

This volume contains discussions of the CAMP parts which are intended to supplement the more detailed descriptions of the CAMP parts provided in the Software Requirements Specification (SRS) and the Top-Level Design Document (TLDD). Each individual CAMP part is not discussed in this volume, rather, part categories are discussed. These discussions describe the motiviation for the parts, background material for understanding the parts, and the applicability of the parts.

Appendix A contains a listing of the CAMP parts identified during the first phase of the CAMP program. It should be noted that we believe more parts will be identified as work proceeds during the second phase of CAMP.

SECTION II
DATA PACKAGE PARTS RATIONALE

1. OVERVIEW

Data packages are parts which encapsulate cohesive sets of data types or constants. A data type is an aggregation of data attributes which characterizes the values that a data object may assume, and the operations which can be performed on the data object. A data constant is a data object with a fixed value.

The pre-definition and pre-packaging of data types and constants are advantageous for both software reliability and productivity. Software will be more reliable with the use of data packages because programmers will be less likely to introduce errors caused by the misuse of data (e.g., mixing different data types such as feet and meters, using the wrong value for a constant, etc.). Software productivity will be enhanced with the use of these data packages because the programmer will not be required to build his own types and constants, and the amount of rework required due to data misuse errors will be reduced.

2. DATA TYPE PACKAGES

Data Type Packages are parts which encapsulate cohesive sets of data types. Because Ada is a strongly typed language, it facilitates the early detection of data misuse errors. For example, the code segment in Figure 1 depicts the calculation of the elevation proportional acceleration command for the BGM-109C missile.

```
ELEVATION_PROPORTIONAL_ACCEL_CMD
     := (-1/RADIUS_OF_CURVATURE) * (EARTH_RELATIVE_VELOCITY**2)
      * (FPS2_TO_SIX_G);
```

Figure 1. Code Segment

The operation shown in Figure 1 could be encapsulated as a function, where RADIUS_OF_CURVATURE and EARTH_RELATIVE_VELOCITY are supplied as parameters. Figure 2 depicts this situation.

```
procedure ACCEL_CMD is
    type FEET is new Float;
    type METERS is new Float;
    subtype FEET_PER_SECOND is Float;
    subtype METERS_PER_SECOND is Float;
    subtype SIX_G is Float;
    RADIUS_OF_CURVATURE : METERS;
    EARTH_RELATIVE_VELOCITY : Feet_Per_Second;
    ELEVATION_PROPORTIONAL_ACCEL_CMD : Six_G;
    function COMPUTER_ACCEL_CMD (RADIUS· FEET;
                                 VELOCITY: FEET_PER_SECOND)
            return SIX_G is separate;
begin
    ELEVATION_PROPORTIONAL_ACCEL_CMD :=
      COMPUTED_ACCEL_CMD (RADIUS  => RADIUS_OF_CURVATURE,
                          VELOCITY => EARTH_RELATIVE_VELOCITY);
end ACCEL_CMD;
```

Figure 2. The Use of the Function

In a strongly typed language such as Ada, each data object (i.e., variable) must be assigned a data type which governs the values which it can take and the operations which can be performed upon it. Figure 2 shows how this would be done for our example. It should be noted here that many of the data type definitions given in the above examples would be supplied to the programmer in standard data type parts.

The Ada compiler will check to make sure that the types of the actual parameters conform to the formal parameters of the function specification. In a non-data typing language such as FORTRAN, subprogram interfaces are not checked at compile time. An error in such an interface would not be detected until the program was executed and an observation was made, hopefully, that the results were incorrect.

The example shown in Figure 2 depicts an error being made in the function call; the programmer has used an object (q.e. RADIUS_OF_CURVATURE) of type METERS where one of FEET has been specified. The Ada compiler will flag this error and therefore allow its early detection and correction. Figure 3 shows the corrected version of the code shown in Figure 2.

The CAMP data type package parts provide a predefined set of data type definitions useful in different components of a missile software system. One example of a data type package which is used often within the CAMP parts is the Basic Units data type package. This part will provide the programmer with predefined data types for basic units used extensively in missile software systems. Figure 4 lists some of the types which are defined in this package.

In addition to defining the data types, these data type packages provide a set of overloaded operator for the encapsulated data types. For example, in the basic units part the "*" operator would be overloaded to tell the Ada compilation system that FEET * FEET yields an object of type FEET.

```
procedure ACCEL_CMD is
   type FEET is new Float;
   type METERS is new Float;
   subtype FEET_PER_SECOND is Float;
   subtype METERS_PER_SECOND is Float;
   subtype SIX_G is Float;
   RADIUS_OF_CURVATURE : FEET;
   EARTH_RELATIVE_VELOCITY : FEET_PER_SECOND;
   ELEVATION_PROPORTIONAL_ACCEL_CMD : SIX_G;
   function COMPUTED_ACCEL_CMD (RADIUS: FEET;
                                VELOCITY: FEET_PER_SECOND)
            return SIX_G is separate;
begin
   ELEVATION_PROPORTIONAL_ACCEL_CMD :=
        COMPUTED_ACCEL_CMD (RADIUS   =>RADIUS_OF_CURVATURE,
                            VELOCITY =>EARTH_RELATIVE_VELOCITY);
end ACCEL_CMD;
```

Figure 3. The Corrected Code Segment

Utility	Units to be Defined
Time:	SECOND, MINUTES
Temperature:	DEGREES_CENTRIGRADE, DEGREES_KELVIN
	DEGREES_FAHRENHEIT
Angle:	DEGREES, RADIANS, SEMICIRCLES, REVOLUTIONS
Distance:	FEET, CENTIMETERS, METERS
Velocity:	FEET_PER_SECOND, CENTIMETERS_PER_SECOND,
	METERS_PER_SECOND, REVOLUTIONS_PER_SECOND
Angular Velocity:	RADIANS_PER_SECOND, DEGREES_PER_SECOND
Acceleration:	FEET_PER_SECOND2, CENTIMETERS_PER_SECOND2,
	METERS_PER_SECOND2, GS
Angular Acceleration:	RADIANS_PER_SECOND2, DEGREES_PER_SECOND2,
	SEMICIRCLES_PER_SECOND2
Distance Variances:	FEET2, CENTIMETERS2, METERS2
Angular Variance:	RADIANS2, DEGREE2,SEMICIRCLE2
Weight:	POUNDS, KILOGRAMS
Voltage:	VOLTS
Pressure:	POUNDS_PER_FEET2, KILOGRAMS_PER_METER2

Figure 4. Types Defined in the Basic Units Part

3. DATA CONSTANT PACKAGES

Data Constant Packages are parts which encapsulate cohesive sets of data constants. The use of this type of part serves to increase both software reliability and software productivity. Software reliability is increased because the use of standard values is enforced. For example, it was observed in the CAMP domain analysis that several different values for the earth equatorial radius were used in the missile software systems studied. The use of a standard part containing earth data will avoid these types of discrepancies. Software productivity is enhanced because the programmer will not have to recreate constants and because errors will be avoided which might require reworking the program.

A good example of a data constant package part is the CAMP part which provides all the data concerning the WGS-72 model of the earth. This package defines a large number of constants which describe various geophysical properties of the earth (see Figure 5).

EARTH_EQUATORIAL_RADIUS	EARTH_POLAR_RADIUS
MEAN_EARTH_RADIUS	EARTH_ECCENTRICITY
EARTH_ELLIPTICITY	EQUATORIAL_GRAVITY
GRAVITY_CORRECTION	EARTH_ROTATION_RATE
EARTH_FLATTENING	INVERSE_EARTH_EAST_RADIUS
INVERSE_EARTH_NORTH_RADIUS	SCHULER_CONSTANTS

Figure 5. Earth Constants

Another example of such a part is the CAMP part which provides the programmer with standard conversion factors. The factors defined in this part are shown in Figure 6.

Feet to centimeters	Feet to meters
Centimeter to Feet	Meters to feet
Gravity (ft/sec2) to metric	Radians to degrees
Degrees to radians	Semicircles to degrees
Degrees to semicircles	Pounds to kilograms
Fahrenheit to celsius	Fahrenheit to kelvin
Kilograms to pounds	Celsius to fahrenheit
Kelvin to fahrenheit	

Figure 6. Conversion Factors

SECTION III
EQUIPMENT INTERFACE PARTS RATIONALE

An equipment interface part standardizes the manner in which the application software interfaces with external hardware peripherals. Although missile systems usually have specialized hardware peripherals, many times a single abstract model of several similar hardware devices (e.g., radar altimeters) can be developed. When this can be achieved, a part can be developed which contains an Ada package specification for the abstract device. The individual differences between devices would be accounted for in Ada package bodies. These package bodies could be parts if a particular device was frequently used within applications. Otherwise they would be developed for each application. In this fashion, the differences between the particular devices can be hidden from the application program. This situation is depicted in Figure 7 for radar altimeters.

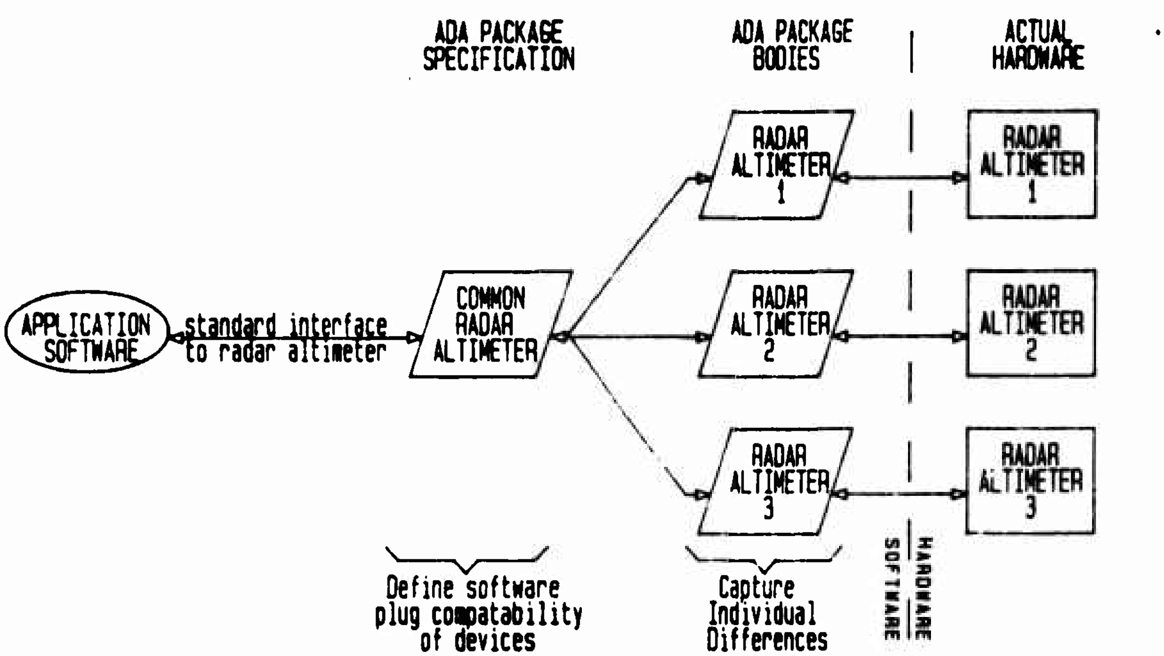

Figure 7. The Equipment Interface Part Scheme

SECTION IV
NAVIGATION PARTS RATIONALE

1. OVERVIEW

The navigation category contains parts which determine the velocity and position of a missile which uses an inertial navigation system. The CAMP domain analysis identified the navigation system as being very amenable to the use of software parts. Although there does exist major areas of variance within the navigation operations, such as the type of navigation coordinate system being used, these sources of variance can be accounted for.

The parts in the navigation category have been bundled into one of three navigation packages based on the dependence of the bundled parts upon a particular local level navigation coordinate system.

-- Common Navigation Bundle (Part R188)
-- Wander Azimuth Navigation Bundle (Part R015)
-- North Pointing Navigation Bundle (Part R016)

These bundles, which are parts in and of themselves, provide the user with a convenient method of accessing the bundled parts along with establishing an appropriate environment for the bundled parts.

2. THE COMMON NAVIGATION BUNDLE

This part contains all the navigation parts which are independent of the type of local level navigation coordinate system. In other words, these parts can be used in either the wander azimuth or north pointing navigation coordinate system environment.

3. THE WANDER AZIMUTH NAVIGATION BUNDLE

This part contains all the navigation parts which are dependent upon the use of a wander azimuth local level navigation coordinate system. A wander azimuth coordinate system is one in which:

-- one of the axes (typically Y) is contained in the tangent and rotated counter-clockwise from north by some wander azimuth angle,
-- another axis (typically Z) is along the gravity vector (typically positive in the up direction), and
-- the third axis (typically X) completes the right-handed orthogonal set.

In this scheme when the wander azimuth angle is zero, X is positive in the East director, Y is positive in the North direction, and Z is positive in the Up direction.

4. THE NORTH POINTING NAVIGATION BUNDLE

This part bundle contains all the navigation parts which are dependent upon the use of a north pointing local level navigation coordinate system. A north pointing coordinate system is one in which:

-- one of the axes (typically X) points East,
-- another axis (typically Y) points North, and
-- the third axis (typically Z) points Up.

SECTION V

KALMAN FILTER PARTS RATIONALE

1. INTRODUCTION

In a missile software system, a Kalman filter is used to combine external measurements of position or velocity with internal calculations of position or velocity. This combination as performed by a Kalman filter weighs the uncertainty of the external measurements against the uncertainty of the internal calculations to produce an optimal estimate of missile position and velocity. In addition the filter produces optimal estimates of inertial sensor errors such as gyro and accelerometer biases.

The 13 software parts shown in Figure 8 have been identified to help the software engineer implement the Kalman filter function.

2. KALMAN FILTER DESCRIPTION

A Kalman filter structure that represents the common top level architecture found in each of the missiles in the CAMP missile set, that have a Kalman filter, is shown in Figure 9. The following discussion is keyed to the block numbers in this figure and serves to introduce terminology.

Part R145 - Propagate State Transition and Process Noise Matrix

Part R146 - Propagate Error Covariance Matrix

Part R147 - Kalman Update

Part R181 - Kalman Update (Complicated H Version)

Part R148 - Propagate State Transition Matrix

Part R149 - Compute Kalman Gain

Part R182 - Compute Kalman Gain (Complicated H Version)

Part R150 - Update Error Covariance Matrix

Part R183 - Update Error Covariance Matrix (Complicated H Version)

Part R151 - Update State Vector

Part R184 - Update State Vector (Complicated H Version)

Part R152 - Sequentially Update Covariance Matrix and State Vector

Part R201 - Sequentially Update Covariance Matrix and State Vector (Complicated H Version)

Figure 8. Kalman Filter Parts

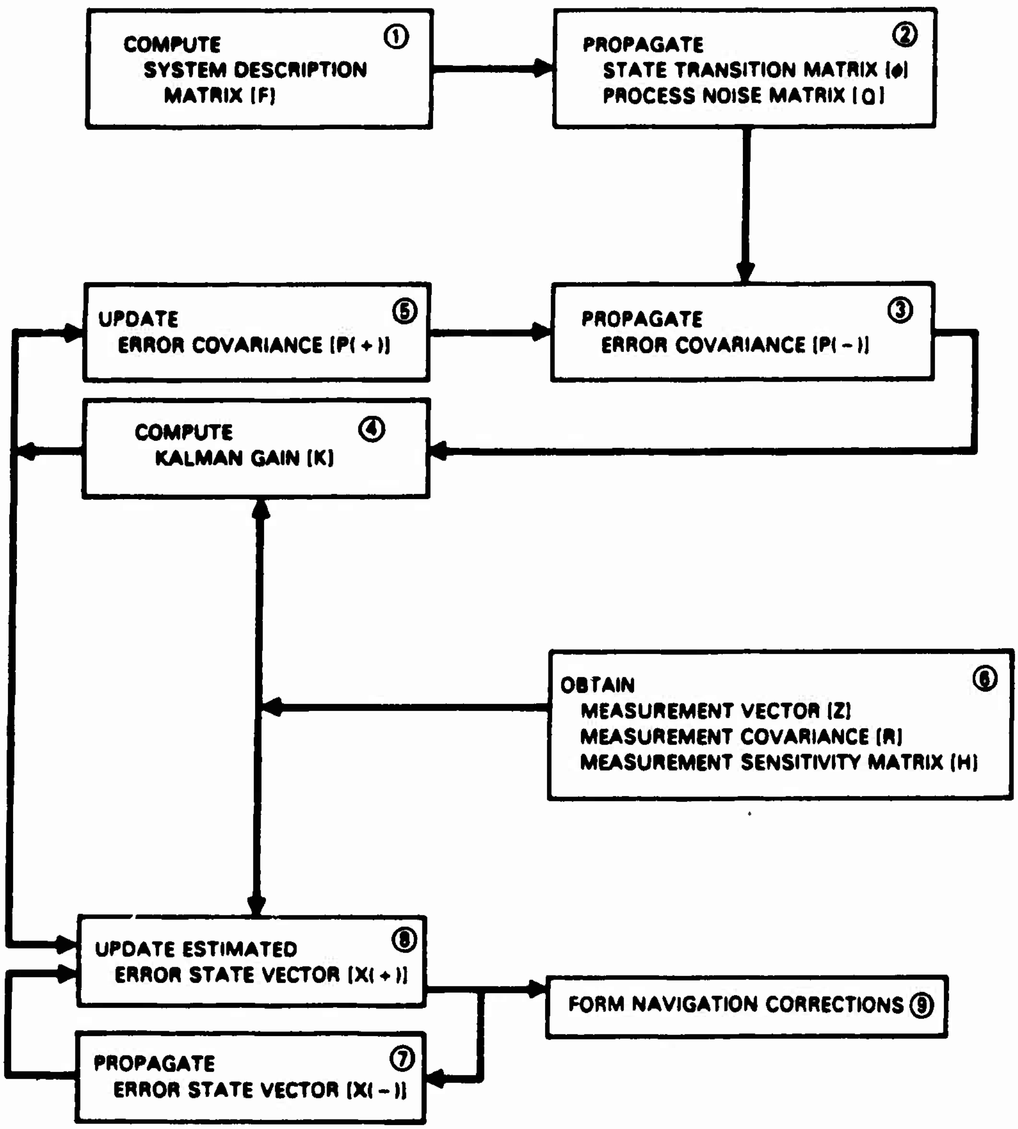

Figure 9. The Common Kalman Filter Model

Block 1

The system description matrix (F) describes the system error dynamics and is a function of time:

$$dX(t)/dt = F(t)*X(t)$$

Here X(t) is the state vector whose components are the errors to be estimated. F(t) is computed periodically by the Kalman filter function based on inputs from the navigation function such as gravity, accelerations, and platform rates.

Block 2

The state transition matrix (PHI) is used to transition the state vector from one instant of time to the next:

$$X(t_k) = PHI(t_k,t_1)*X(t_1)$$

PHI is propagated according to:

$$PHI(t_k,t_1)=PHI(t_k,t_{k-1})*PHI(t_{k-1},t_1)$$

$PHI(t_k,t_{k-1})$ is approximated using $F(t_{k-1})$ computed in Block 1.

The process noise matrix (Q) is a measure of the noise driving the error dynamics. It can also be propagated from one instant of time to the next.

Block 3

P(-) is the covariance matrix of the state vector. It is a measure of the uncertainty in the state vector. P(-) is propagated from one instant of time to the next using the state transition matrix (PHI) and the noise (Q) that drives the process.

Block 4

The Kalman gain (K) is computed, when a measurement is available, by an expression which considers both the uncertainty in the state, quantified by P(-), and the uncertainty in the measurement, quantified by R.

Block 5

Once a measurement update is available, the uncertainty in the state vector (X) decreases. This is reflected by a change in the state covariance matrix which is transformed from P(-) to P(+). The "+" indicates the covariance matrix just after a measurement update, whereas "-" indicates the covariance matrix just before a measurement update.

Block 6

The following are typical of systems from which external measurements are obtained.

- -- TERCOM: TERrain COrrelation Matching provides Lat/Long measurements using a radar altimeter.
- -- DSMAC: Digital Scene Matching Area Correlation provides Lat/Long measurements using a passive imaging device in the visible spectrum.
- -- Radar Altimeter
- -- Barometric Altimeter
- -- GPS: Global Positioning System obtains position and/or velocity measurements by communication with an orbiting satellite network.
- -- Platform Navigator: Prior to launch, position and velocity measurements can be obtained from the launch platform.

The external measurement vector (Z) is related to the state vector (X) by the measurement sensitivity matrix (H): Z = H*X

Associated with each measurement source is a measure of the uncertainty in the measurement--the measurement covariance matrix (R).

Block 7

The Kalman filter state vector (X) consists of components which are the estimates of errors in navigation parameters and in inertial sensor characteristics. These error estimates are used to correct navigation position and velocity, and also to correct the inertial sensors for bias, scale factor error, etc. The state vector is propagated from one instant to the next by use of the state transition matrix as discussed in Block 2.

Block 8

Once a measurement is available, the Kalman gain is computed and the state vector can be updated:

$$X(+) = X(-)+K*[\ Z-H*X(-)\ ]$$

The "+" indicates the state vector just after a measurement update whereas "-" indicates the state vector just prior to the measurement update.

Block 9

After an updated state vector is available, processing may be required to transform the state vector components representing navigation errors into a form suitable for navigation corrections. A simple example of this is the conversion of position error from units of feet to units of radians.

3. KALMAN FILTER OPERATION

The Kalman Filter parts can be organized hierarchically as shown in Figure 10. Part R145 and R147 are at the top of the hierarchy and are the key parts required to do the core of the Kalman filter processing task. As indicated in the figure, these key parts draw upon the resources provided by the lower level parts: R152, R148, R149, R150, R151, and R146.

The operation of the two key parts, R145 and R147, is shown in Figure 11. As seen in the figure, part R145 processes measurements, Z_1 through Z_N, which occur at times t_1 through t_N, to form a measurement information file containing records of the form: (Z, H, R, PHI, Q). This is discussed in the following paragraphs.

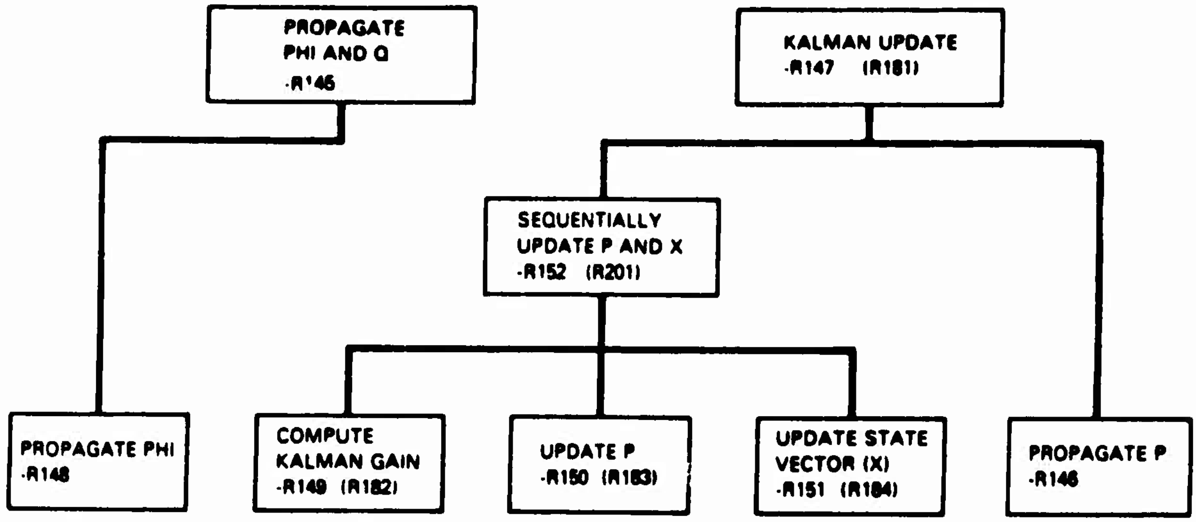

Figure 10. Kalman Filter Functional Part Hierarchy

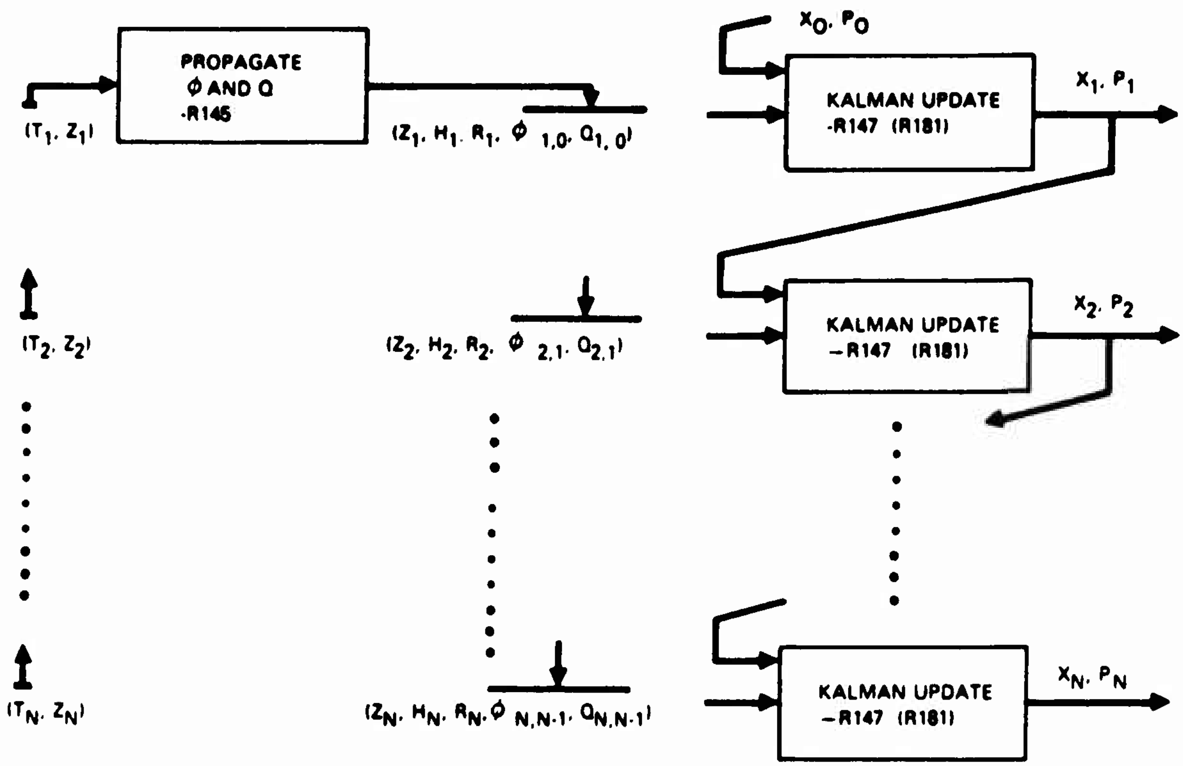

Figure 11. Key Kalman Filter Part Operation

When time t_1 is equal to the real time, part R145 is used to store a propagated Phi and a propagated Q. These two propagated quantities, along with the measurement (Z_1) and the measurement sensitivity and covariance matrices (H and R) are stored as a record in the measurement information file. This record is shown as (Z_1, H_1, R_1, $PHI_{1,0}$, $Q_{1,0}$) in the figure. Likewise, when times t_2 through t_N arise the same process described above is repeated for each.

Note that only the times t_1 through t_N are required in real time. The measurements Z_1 through Z_N can be inserted into the appropriate record on a delayed basis relative to real time. Because of this, the algorithm is said to be capable of processing delayed observables.

As soon as a record is present in the measurement information file, part R147 can be used to process the record. Using the information in the record, along with the previous state vector estimate and corresponding covariance matrix, this part computes a new state vector estimate and its corresponding covariance matrix.

After the measurement information record corresponding to the last measurement, Z_N, has been processed by part R147, the components of the state vector, X_N, are optimal estimates of navigation system errors and are valid at time, t_N. To propagate these errors forward from time, t_N, to the current time, t_C, part R145 is requested to supply the state transition matrix, Phi(t_C,t_N). After multiplying X_N by Phi(t_C,t_N) the resulting state error estimate, X_C, is made available to the navigation subsystem for correction of navigation errors.

4. PARTS DESCRIPTION

a. Propagate State Transition and Process Noise Matrix (Part R145)

This part executes each time it is supplied with an integrated system description matrix. Using the integrated system description matrix it propagates the state transition matrix (Phi) and the process noise matrix (Q) forward in time. Both the Phi and Q matrices span a time interval ranging from the point in time that a measurement was last processed to the current real time. At the time of a new measurement this part, upon request,

supplies the Phi and Q matrices that it has been propagating. The suppliedmatrices span a time interval from the previous measurement time to the new measurement time. After suppling these matrices they are reset - marking the beginning of another time interval.

At the time that a state vector is to be used to correct navigation system errors, this part is requested to supply only the Phi matrix. In this case Phi spans the time interval from the time of the last measurement to the current real time (i.e., the time the navigation corrections are to be applied).

b. Propagate Error Covariance Matrix (Part R146)

This part uses the propagated state transition matrix, Phi, and the propagated process noise matrix, Q, to propagate the error covariance matrix, P, forward to the time of the latest measurement.

c. Kalman Update (Part R147)

This part produces an optimal estimate of the state vector (i.e., navigation errors) given the following measurement information:

-- the measurement vector, Z,
-- the measurement sensitivity matrix, H,
-- the measurement covariance matrix, R, and
-- the state transition matrix, Phi, and the process noise matrix, Q, which have been propagated to span the time interval from the last measurement to the time of the present measurement.

This part also computes the error covariance matrix, P, which quantifies the uncertainty present in the state vector estimate.

In performing the above functions, this part uses the lower level parts, R146 and R152.

The use of part R147 is appropriate only if the following two assumptions are satisfied:

- -- Any row of the measurement sensitivity matrix, H, contains only one "1" with the other row elements being "0". In other words, any measurement component is a direct measurement of a single component of the state vector.
- -- The measurement covariance matrix, R, is diagonal.

d. Kalman Update - Complicated H Version (Part R181)

This part is the same as part R147 with the exception that the assumption regarding the measurement sensitivity matrix, H, is replaced by the assumption that H is sparse. This assumption allows the relationship between a measurement component and the components in the state vector to be more complicated.

e. Propagate State Transition Matrix (Part R148)

This part is used to compute a state transition matrix that spans time interval which is a multiple of the basic time interval used for integration of the system description matrix.

f. Compute Kalman Gain (Part R149)

This part computes the Kalman gain (a vector in this case) for a single measurement component.

The use of part R149 is appropriate only if the following two assumptions are satisfied:

- -- Any row of the measurement sensitivity matrix, H, contains only one "1" with the other row elements being "0". In other words, any measurement component is a direct measurement of a single component of the state vector.
- -- The measurement covariance matrix, R, is diagonal.

g. Compute Kalman Gain - Complicated H Version (Part R182)

This part is the same as part R149 with the exception that the assumption regarding the measurement sensitivity matrix, H, is replaced by the assumption that H is sparse. This assumption allows the relationship between a measurement component and the components in the state vector to be more complicated.

h. Update Error Covariance Matrix (Part R150)

This part computes P(+), the uncertainty in the state vector after a measurement is made. For this computation the part uses:

-- P(-), the uncertainty in the state vector before the measurement,
-- the Kalman gain, and
-- the measurement sensitivity matrix, H.

The use of part R150 is appropriate only if the following two assumptions are satisfied:

-- Any row of the measurement sensitivity matrix, H, contains only one "1" with the other row elements being "0". In other words, any
-- measurement component is a direct measurement of a single component of the state vector.
-- The measurement covariance matrix, R, is diagonal.

i. Update Error Covariance Matrix - Complicated H Version (Part R183)

This part is the same as part R150 with the exception that the assumption regarding the measurement sensitivity matrix, H, is replaced by the assumption that H is sparse. This assumption allows the relationship between a measurement component and the components in the state vector to be more complicated.

j. Update State Vector (Part R151)

This part computes the optimal estimate of the state vector, X(+), from:

-- X(-), the state vector before incorporation of the information available in the measurement,
-- the Kalman gain,
-- the measurement sensitivity matrix, H, and
-- the measurement, Z.

The use of part R151 is appropriate only if the following two assumptions are satisfied:

-- Any row of the measurement sensitivity matrix, H, contains only one "1" with the other row elements being "0". In other words, any measurement component is a direct measurement of a single component of the state vector.
-- The measurement covariance matrix, R, is diagonal.

k. Update State Vector - Complicated H Version (Part R184)

This part is the same as part R151 with the exception that the assumption regarding the measurement sensitivity matrix, H, is replaced by the assumption that H is sparse. This assumption allows the relationship between a measurement component and the components in the state vector to be more complicated.

l. Sequentially Update Covariance Matrix and State Vector (Part R152)

This part computes an updated state vector, X(+), and the updated error covariance matrix, P(+), by using the lower level parts: R149, R150, and R151.

In doing this, the components of the measurement vector are treated as a sequence of simultaneous scalar measurements which are processed

in turn by the three parts R149, R150, and R151. This approach avoids the computation of a matrix inverse that would otherwise be necessary in the error covariance matrix update operation.

The use of part R152 is appropriate only if the following two assumptions are satisfied:

-- Any row of the measurement sensitivity matrix, H, contains only one "1" with the other row elements being "0". In other words, any measurement component is a direct measurement of a single component of the state vector.
-- The measurement covariance matrix, R, is diagonal.

m. Sequentially Update Covariance Matrix and State Vector - Complicated H Version (Part R201)

This part is the same as part R152 with the exception that the assumption regarding the measurement sensitivity matrix, H, is replaced by the assumption that H is sparse. This assumption allows the relationship between a measurement component and the components in the state vector to be more complicated.

5. PARTS USAGE

All the Kalman filter parts discussed above, with the exception of the "complicated H versions", could have been used in seven of the ten missiles in the CAMP missile set. This is shown in Figure 12.

The complicated H version parts (R181, R182, R183, and R184), are intended for use with missiles whose external measurement subsystems measure linear combinations of state vector components. An example of this occurs on the cruise missile advanced guidance program which uses a velocimeter that measures line of sight velocity along an axis fixed relative to the vehicle body.

A: UTG SINP B: AGM-109H C: AGM-109L D: BGM-109C E: BGM-109B

F: BGM-109G G: HARPOON H: BGM-109A I: MGD GANP J: SPARTAN

	A	B	C	D	E	F	G	H	I	J
Kalman Filter Parts (except for complicated H versions).....	X	X	X	X		X		X	X	

Figure 12. Kalman Filter Parts Usage

SECTION VI
WAYPOINT STEERING PARTS RATIONALE

1. INTRODUCTION

Waypoint steering software is used in missiles that have an inertial navigation system to control the missile's ground track. Given navigated missile latitude and longitude, waypoint steering software computes missile crosstrack and heading error relative to the desired missile ground track (contained in pre-stored mission data). The crosstrack and heading error are then used by the lateral guidance function to generate a roll command for the lateral/directional autopilot. The lateral/directional autopilot causes the missile to roll until the crosstrack and heading error are nulled.

The 13 software parts shown in Figure 13 have been identified to help the software engineer implement the waypoint steering function.

2. GENERAL DESCRIPTION

At any given time three waypoints determine the missile ground track. This is illustrated in Figure 14 for the case of nonturning flight. The missile flies away from the past waypoint (point A) towards the current waypoint (point B). The flight path is along the current course segment (segment AB). Point C is referred to as the next waypoint. The current course segment and the next course segment (segment BC), along with the specified turn radius, determine the turn angle and the turning distance shown in Figure 14.

Part R168 - Compute Unit Radial Vector

Part R169 - Compute Segment Unit Normal Vector

Part R170 - Initialize Steering Vectors

Part R171 - Update Steering Vectors

Part R172 - Compute Turn Angle and Direction

Part R180 - Perform Start/Stop Turn Test

Part R179 - Perform Start Turn Test

Part R178 - Perform Stop Turn Test

Part R174 - Compute Crosstrack and Heading Error

Part R173 - Compute Crosstrack and Heading Error When Turning

Part R175 - Compute Crosstrack and Heading Error When Not Turning

Part R176 - Compute Distance to Current Waypoint

Part R177 - Compute Turning and Nonturning Distances

Figure 13. Waypoint Steering Parts

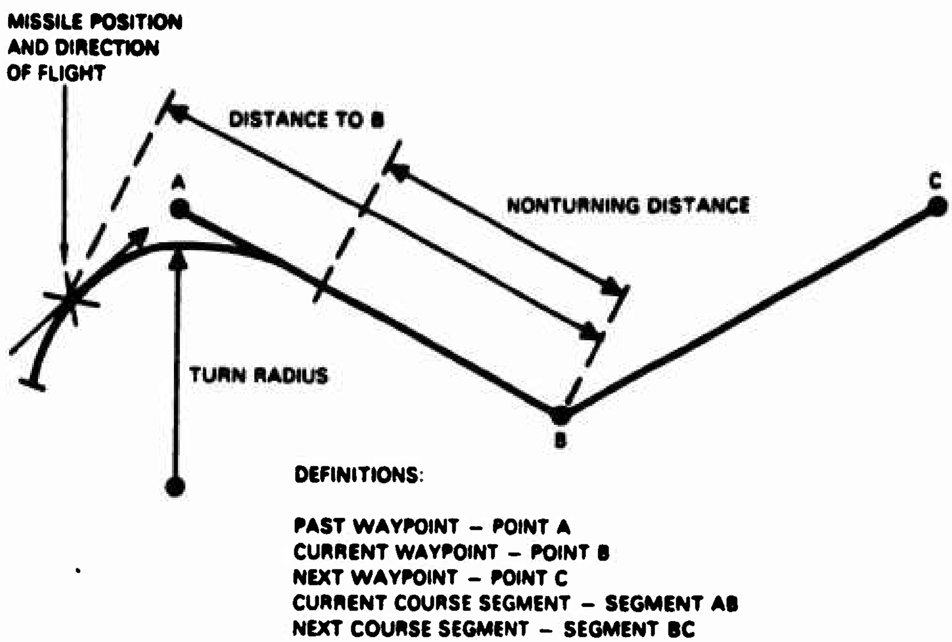

Figure 14. Non-turning Waypoint Steering

When the distance to B is equal to the turning distance (adjusted by a lead distance) the missile initiates a turn. At this time a new next waypoint is obtained from pre-stored mission data. The waypoints A, B, and C are then redefined as shown in Figure 15. After redefinition the prior waypoints B and C become waypoints A and B. The missile now follows a constant radius turn circle as shown in the figure.

The turn is stopped when the distance to B is equal to the nonturning distance (adjusted by a lead distance). At this time, the missile again flies in nonturning flight along the current course segment and the steering process is ready to repeat.

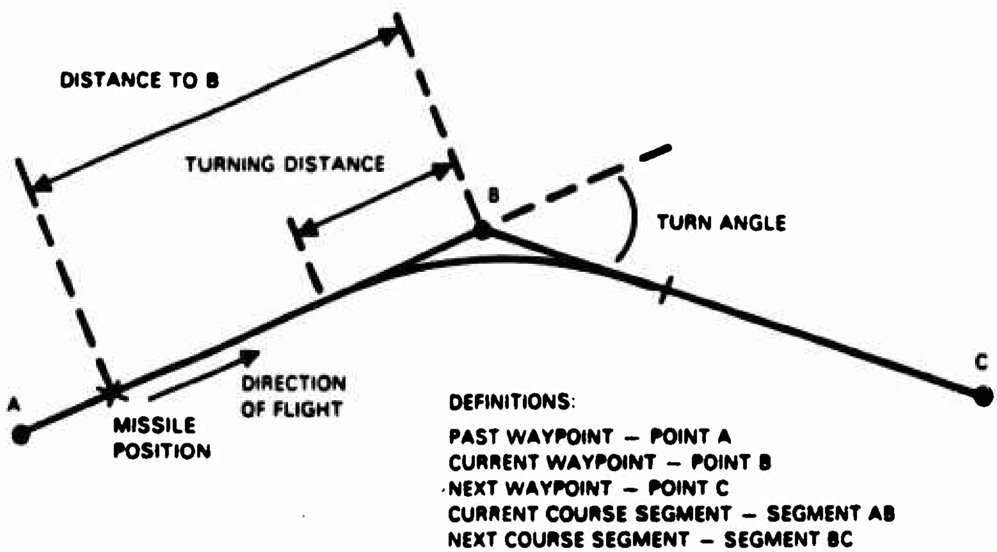

Figure 15. Turning Waypoint Steering

3. DETAILED OPERATION

The operation of the waypoint steering software function can be expressed as a finite state machine with three states: quiescent, not-in-turn, and in-turn. Figure 16 shows, for each state transition, the stimulus above the transition line and the actions (processing) below the transition line.

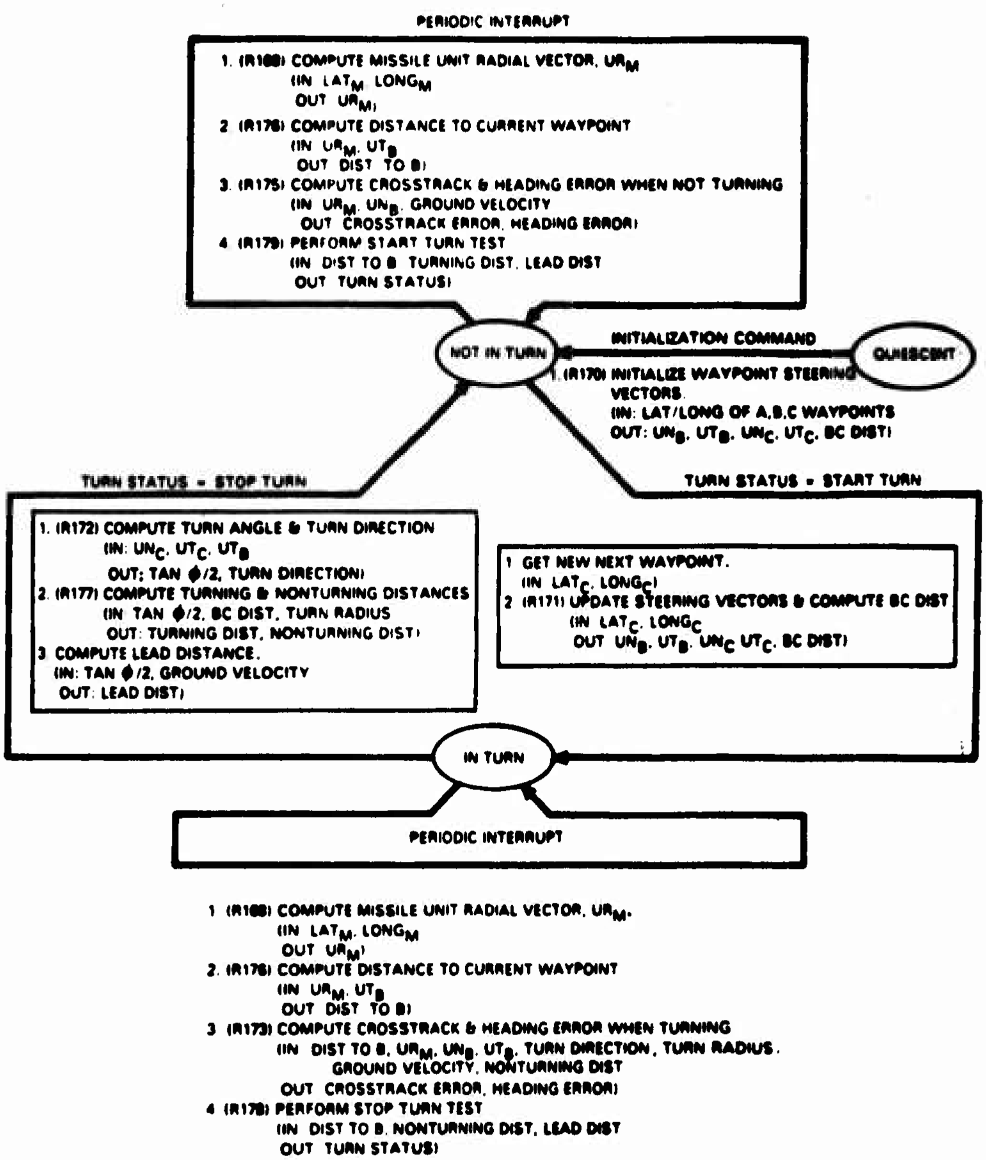

Figure 16. Waypoint Steering State Diagram

The following paragraphs describe the waypoint steering function with respect to this figure.

To begin the waypoint steering operation the not-in-turn state is entered from the quiescent state after receipt of an initialization command stimulus. Part R170 performs the processing associated with this transition. It computes the initial waypoint steering vectors from the latitude and longitude of the initial waypoints A, B, and C.

In the not-in-turn state, parts R168, R176, R175, and R179 execute periodically to:

-- Compute the missile crosstrack error and heading error.
-- Perform a test to see if it is time to begin a turn.

When the start turn criterion is met the software leaves the not-in-turn state and enters the in-turn state. In this transition a new next waypoint is obtained from stored mission data and part R171 updates the steering vectors for the new waypoints.

In the in-turn state, parts R168, R176, R173, and R178 execute periodically to:

-- Compute the missile crosstrack error and heading error.
-- Perform a test to see if it is time to stop the turn.

When the stop turn criterion is met the software leaves the in-turn state and enters the not-in-turn state. The processing performed in this transition is:

-- Part R172 computes the turn angle and the turn direction.
-- Part R177 computes the turning distance and the nonturning distance.
-- A missile peculiar part computes the lead distance - a parameter which compensates for the delay in turning caused by missile dynamics.

4. PARTS DESCRIPTION

a. Compute Unit Radial Vector (Part R168)

This part computes the unit radial vector of a point given the point's geocentric latitude and longitude. The unit radial vector of a point extends outward, from the origin of the earth centered reference frame, towards the point whose latitude and longitude are given. This is illustrated in Figure 17. This part computes the unit radial vector for either a missile or for a waypoint.

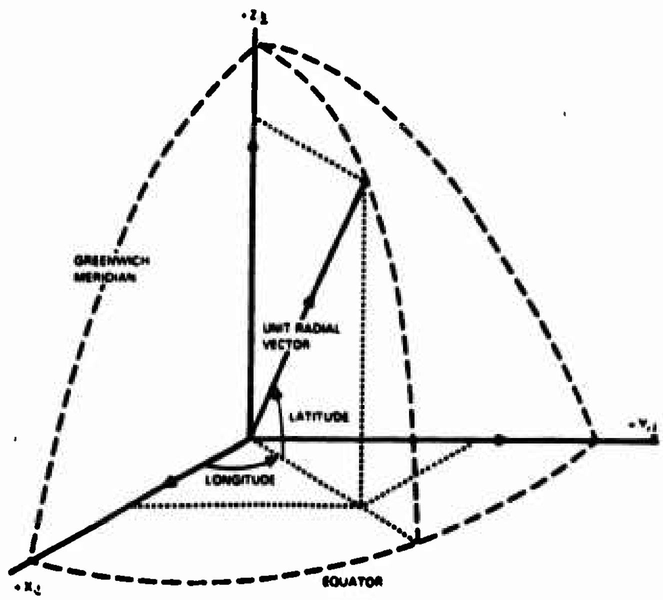

Figure 17. Unit Radial Vector

b. Compute Segment Unit Normal Vector (Part R169)

This part computes the unit normal vector of a course segment from the two unit radial vectors representing the end point positions of the course segment. It also computes the length of the course segment. A unit normal vector is perpendicular to the plane containing the unit radial vectors as shown in Figure 18.

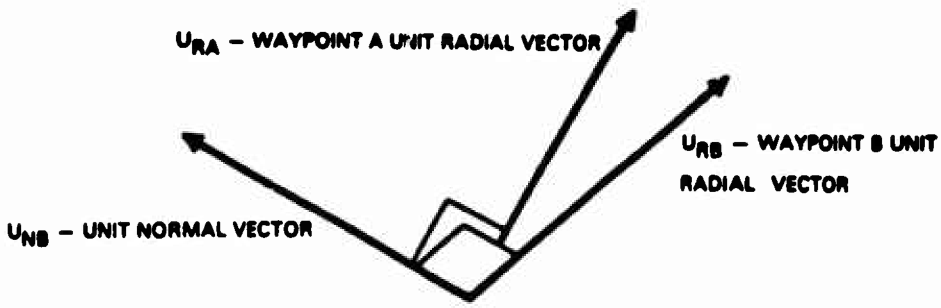

Figure 18. Unit Normal Vector

c. Initialize Steering Vectors (Part R170)

This part performs the initial computation of the steering vectors: UN_B, UT_B, UN_C, and UT_C. It also computes the distance of the next segment which connects waypoints B and C. Figure 19 illustrates the information that this part computes.

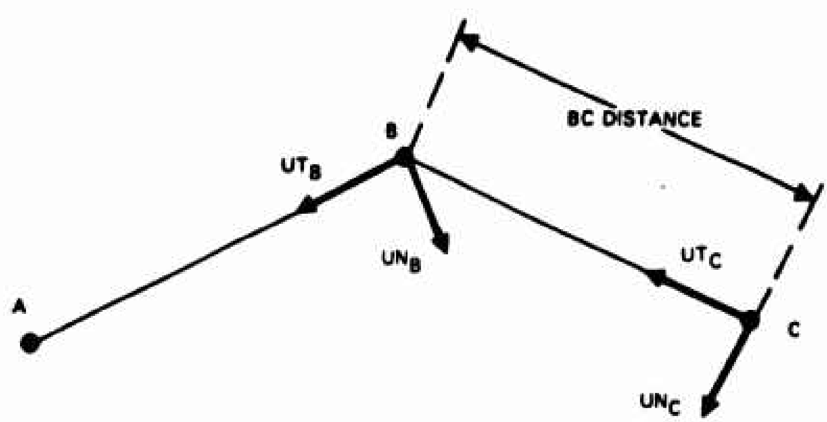

Figure 19. Waypoint Steering Vector Representation

d. Update Steering Vectors (Part R171)

Prior to a waypoint update the missile is in nonturning flight along the current course segment. The waypoint steering vectors are shown for this case in the upper portion of Figure 20. When the missile begins a turn, a new next waypoint (i.e., a new waypoint C) is obtained from mission data. Part R171 then updates the waypoint steering vectors as shown in the lower portion of Figure 20.

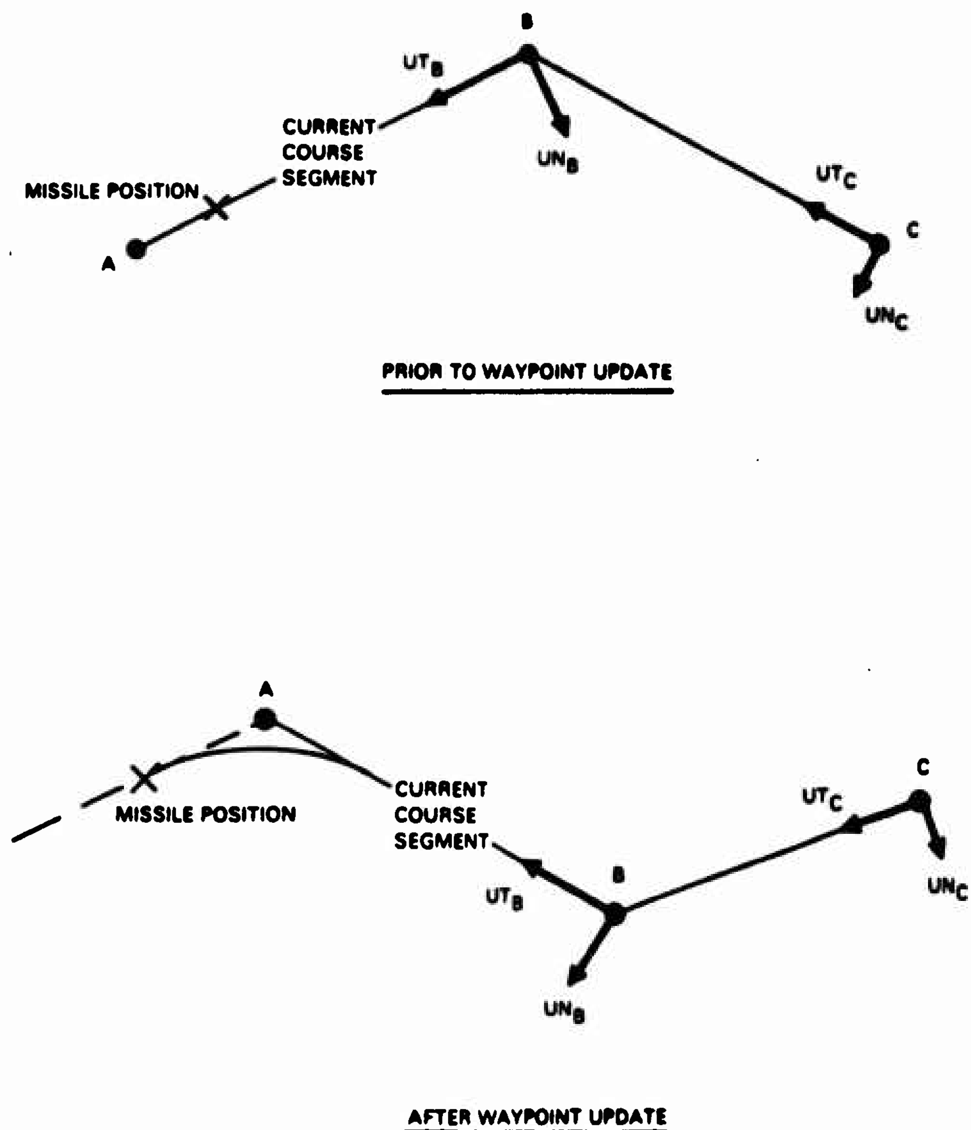

Figure 20. Waypoint and Steering Vector Update

e. Compute Turn Angle and Direction (Part R172)

This part uses the steering vectors (UN_C, UT_C, UT_B) to compute the turn direction and the tangent of one-half the turn angle.

f. Perform Start/Stop Turn Test (Part R180)

This part consolidates the start turn test of Part R179 and the stop turn test of Part R178 in one place.

g. Perform Start Turn Test (Part R179)

This part determines when a turn should start.

h. Perform Stop Turn Test (Part R178)

This part determines when a turn should stop.

i. Compute Crosstrack and Heading Error (Part R174)

This part combines the functions of parts R173 and R175 to reduce the total lines of code and consolidate crosstrack and heading error calculations in one place. See Figure 21 for an illustration of crosstrack and heading error for turning and nonturning flight.

j. Compute Crosstrack and Heading Error When Turning (Part R173)

This part computes crosstrack error and heading error, when the missile is in a turn, as shown in the lower portion of Figure 21.

k. Compute Crosstrack and Heading Error When Not Turning (Part R175)

This part computes crosstrack error and heading error, when the missile is not in a turn, as shown in the upper portion of Figure 21.

Turning distance is illustrated in Figure 14 and nonturning distance in Figure 15.

5. PARTS USAGE

The waypoint steering parts discussed above could have been used in seven of the ten missiles in the CAMP missile set. This is shown in Figure 22. Generally any missile that uses an inertial navigation system to determine latitude and longitude can use the waypoint steering parts, described here, to control missile ground track.

A: UTG SINP	B: AGM-109H	C: AGM-109L	D: BGM-109C	E: BGM-109B
F: BGM-109G	G: HARPOON	H: BGM-109A	I: MGD GANP	J: SPARTAN

	A	B	C	D	E	F	G	H	I	J
Waypoint Steering Parts.........	X	X	X	X		X		X	X	

Figure 22. Waypoint Steering Parts Usage

SECTION VII
AUTOPILOT PARTS RATIONALE

1. INTRODUCTION

A missile autopilot accepts commands from the guidance function and forces the missile dynamics to conform to the commands in a stable fashion. To do this, the autopilot computes missile fin deflections using dynamic feedback such as angular rates and linear accelerations to stabilize short period airframe motion.

The software parts listed in Figure 23 have been identified to help the software engineer implement the autopilot function. Part R059 (Pitch Autopilot) and Part R064 (Lateral/Directional Autopilot) are the key autopilot parts. Together they can be used to implement the autopilot for a bank-to-turn missile. To perform their function they use the lower level part R048 (Integral Plus Proportional (IPP) Gain) as well as signal processing parts such as digital filters and limit functions.

In a skid-to-turn missile the autopilot would be implemented by using Part R059 (Pitch Autopilot) for both the pitch autopilot function as well as the yaw autopilot function. In this case the use of Part R064 (Lateral/Directional Autopilot) would not be required.

Part R048 - Integral Plus Proportional (IPP) Gain
Part R059 - Pitch Autopilot
Part R064 - Lateral/Directional Autopilot

Figure 23. Autopilot Parts

2. PARTS DESCRIPTION

a. Integral Plus Proportional (IPP) Gain (Part R048)

This part implements a classical control system structure called an "integral plus proportional gain". A block diagram representation of this is shown in Figure 24. The part has a single input which is processed by two parallel paths. An integration of the input is performed by one of the paths while the other simply applies a proportional gain to the input. The results of the integration and gain are then added together to form the output of the part.

Other significant features of this part are:

-- The value that the integrator can attain is limited and the integrator is frozen when its output is in the limited condition.

-- The integrator is also frozen based on the value of a flag (labeled "Clamp" in Figure 24) which represents the status of an external limit operation. This prevents the IPP integrator from charging up when the IPP output is being limited externally.

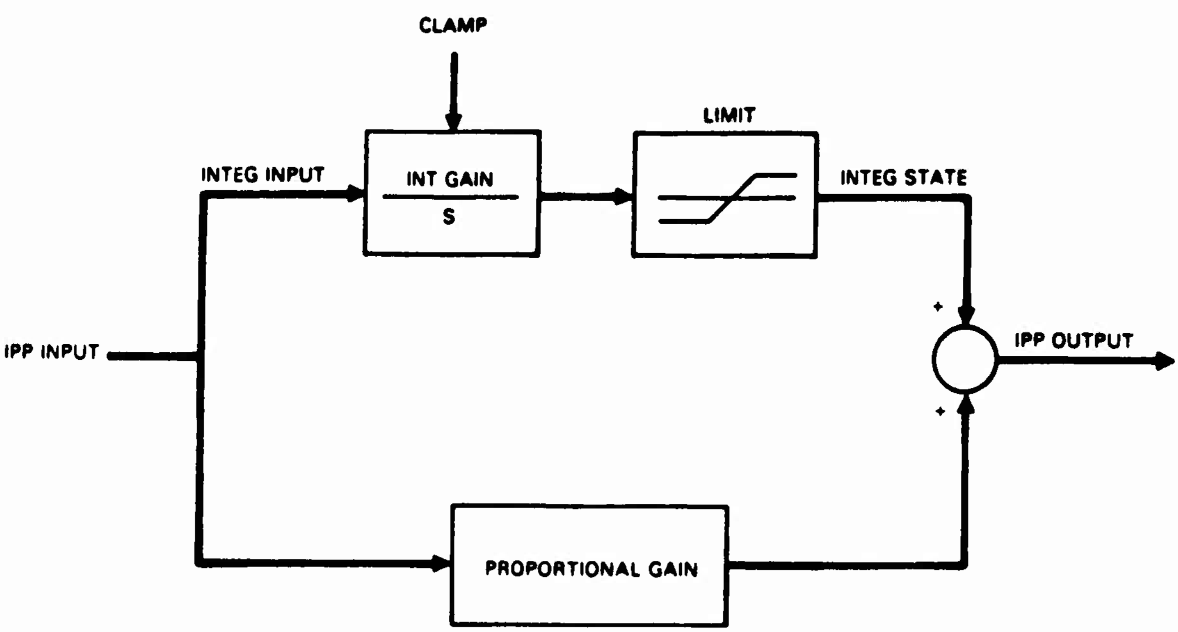

Figure 24. Integral Plus Proportional (IPP) Gain

b. Pitch Autopilot (Part R059)

As shown in Figure 25, this part accepts a normal acceleration command as an input. Then, using normal acceleration and pitch rate feedback, elevator deflection commands are computed in such a manner that missile pitch motion is stable, and actual missile normal acceleration is equal to the commanded normal acceleration.

Digital filters are used to filter the pitch rate and normal acceleration feedback. These are tailored to the particular missile application. For example, in a missile with significant body bending these filters would be selected to prevent body bending modes from adversely affecting vehicle stability. In another missile, body bending may not be a problem so the filters would be removed from the pitch autopilot.

Other significant features of this part are:

-- If the elevator command is on the limit then the integrator on the normal acceleration error signal is frozen.
-- The user is provided with an initialization operation which sets the integrator initial value so that there is no initial elevator movement. This avoids large transient conditions when the pitch autopilot is first activated.

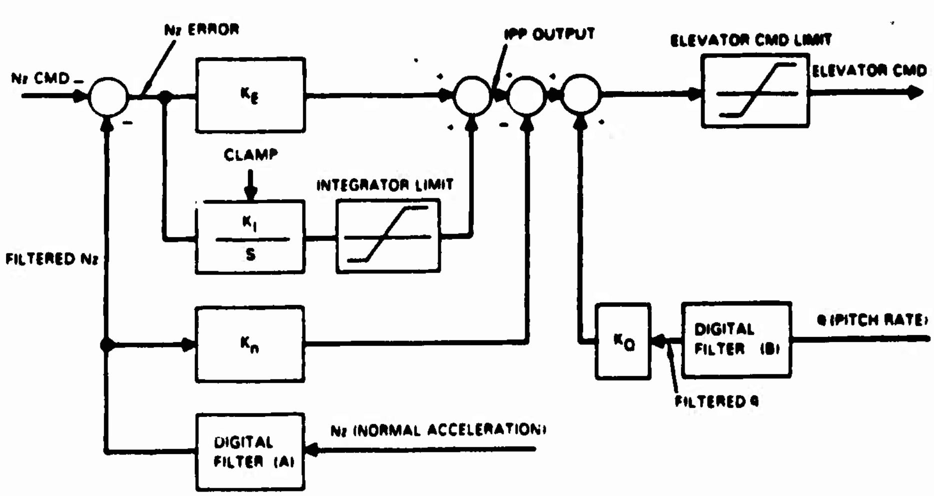

Figure 25. Pitch Autopilot Functional Diagram

c. Lateral/Directional Autopilot (Part R064)

As shown in Figure 26, this part accepts a roll angle command as an input. Then, using roll attitude, lateral acceleration, yaw rate, and roll rate feedback, aileron and rudder deflection commands are computed in such a manner that missile roll and yaw motion are stable, and actual missile roll angle is equal to the commanded roll angle.

Digital filters are used to filter the yaw rate and lateral acceleration feedback. These are tailored to the particular missile application in the same manner as for the pitch autopilot. A digital filter on the roll command is also available, at the option of the application engineer, to prevent step changes in roll command from reaching the aileron command signal.

Other significant features of this part are:

-- If the aileron (or rudder command) is on the limit then the aileron (or rudder) channel integrator is frozen.
-- The user is provided with an initialization operation which sets the initial values of the aileron and rudder channel integrators so that there is no initial aileron or rudder movement. This avoids large transient conditions when the lateral/directional autopilot is first activated.

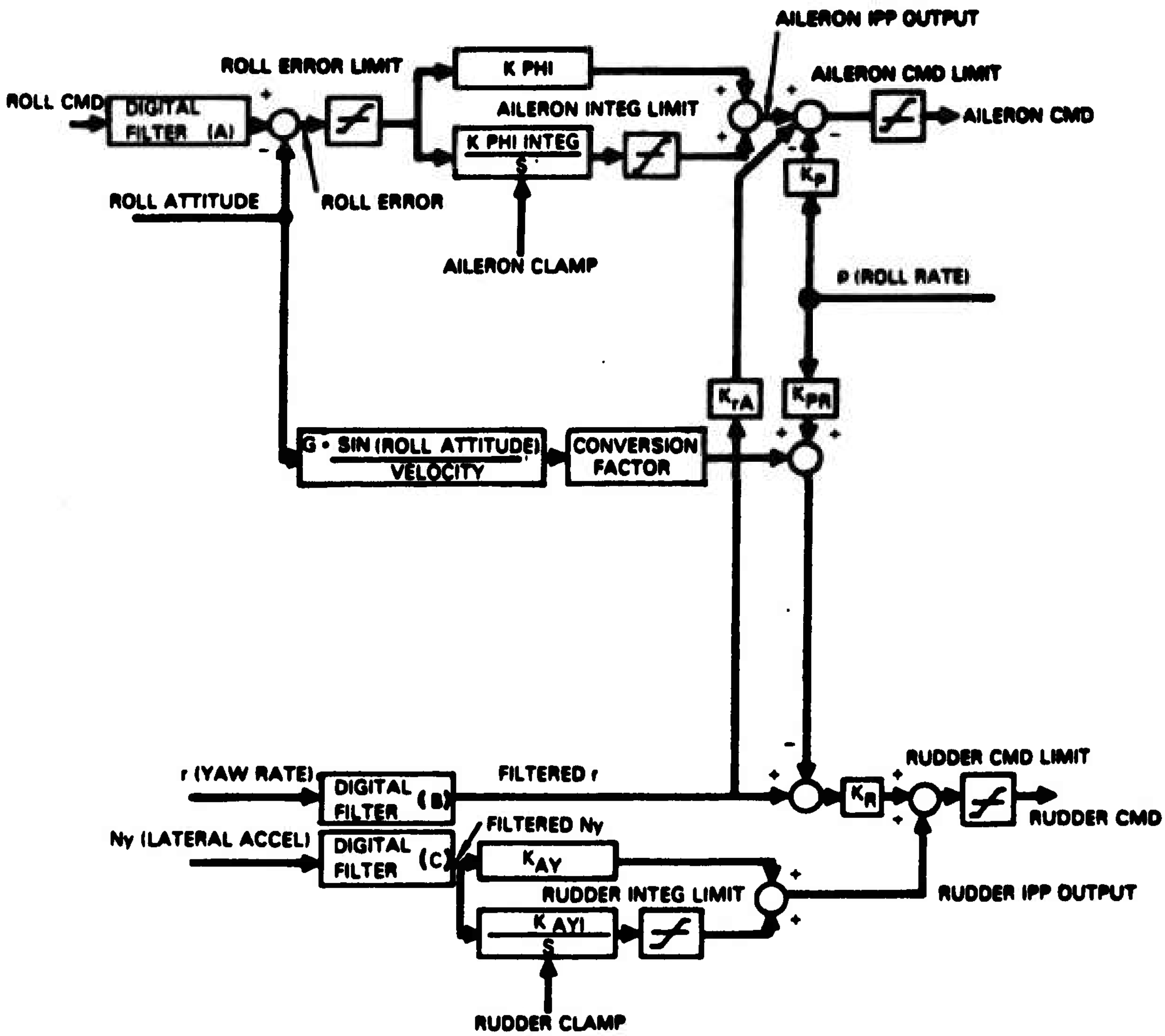

Figure 26. Lateral/Directional Autopilot

3. PARTS USAGE

The application, with respect to the CAMP missile set, of the autopilot parts discussed above is shown in Figure 27. Generally, Part R059 is a candidate for use in any skid-to-turn missile, while the combination of Parts R059 and R064 is a candidate for use in any bank-to-turn missile.

A: UTG SINP B: AGM-109H C: AGM-109L D: BGM-109C E: BGM-109B
F: BGM-109G G: HARPOON H: BGM-109A I: MGD GANP J: SPARTAN

	A	B	C	D	E	F	G	H	I	J
R048 Integral Plus Proportional (IPP) Gain....		X	X	X		X		X		
R059 Pitch Autopilot..............		X	X	X	X	X	X	X		X
R064 Lateral/Directional Autopilot..................		X	X	X		X		X		

Figure 27. Autopilot Part Usage

SECTION VIII
WARHEAD CONTROL COMMONALITY

1. INTRODUCTION

Warheads found in the CAMP missile set can be divided into the following categories:

-- Nuclear (TLAM-N, TLAM-G)
-- Conventional/Programmed Detonation (TLAM-C)
-- Conventional/Impact Detonation (TASM, HARPOON, MRASM-L)
-- Submunition (MRASM-H)

The following paragraphs discuss each warhead category with respect to the common functions that exist in each category, and the reusable software parts that can be used to help implement these functions. A summary of warhead control functions versus the CAMP missiles is shown in Figure 28. Note that UTGV, MGD, and Spartan are not listed in the figure because they do not involve onboard warhead control.

2. NUCLEAR

The missiles in the Camp missile set with nuclear warheads are TLAM-G and TLAM-N. TLAM-G uses the W84 warhead while TLAM-N uses the W80 warhead. Even though the specific warheads are different the following operations were found to be common to the missiles in the nuclear warhead category:

-- Environmental Sensing Device (ESD) Word Formation
-- Warhead Arming Maneuver

WARHEAD CONTROL FUNCTION	MRASM/H	MRASM/L	TLAM/N	TASM	TLAM/C	TLAM/G	HARPOON
ESD Word Formation			x			x	
Warhead Arming Maneuver			x			x	
Function Select Sequencing			x			x	
Final Warhead Arming Sequencing			x			x	
Target Countdown Sequencing			x			x	
Arming Sequence		x		x	x		x
Detonation Time Calculations					x		
Detonation Sequencing					x		
Submunition Calculations	x						

Figure 28. Warhead Control Function Matrix

-- Function Select Sequencing

-- Final Warhead Arming Sequencing

-- Target Count Down Sequencing

The Environmental Sensing Device (ESD) Word Formation function forms a word that contains the results of unique events that occur during a flight. Correct formation of the ESD word, by missile software, is a prerequisite for warhead detonation. An example of an ESD word is given in Figure 29. As seen in the figure, various constraints are placed on each event in relation to the other events. For example, when event 3 (missile enable) occurs it can be recorded in the ESD word only if events 1 and 2 have already occurred, but event 4 has not occurred. The logic to handle the formation of the ESD word can be built using the CAMP abstract process part called a Mealy machine. A Mealy machine, which forms the 24 event ESD word for TLAM-G and TLAM-N, is diagrammed in Figure 30. In this diagram the stimulus for the state transition is shown above the transition line. For example, the transition from state 1 to state 2 occurs when event 1 (E1) has occurred but event 2 (E2) has not. Although not shown in the figure, the processing, associated with a transition from state N, is that event N is recorded in the ESD word.

The Warhead Arming Maneuver involves a sequence of flight maneuvers designed to actuate the warhead ESD acceleration switch. Because this operation is identical in the two missiles, a simple part could be specified for this function.

The Function Select Sequencing operation involves a straightforward sequence of logical tests and timed discrete outputs. The details are different for the two missiles but the pattern of logic is the same.

The Final Warhead Arming Sequencing function involves a straightforward sequence of discrete outputs, including the initiation of the ESD word output. This function is identical for the two missiles and a simple part could be specified to serve the needs of both missiles.

EVENT NO.	EVENT NAME	EVENTS THAT MUST HAVE OCCURRED	EVENTS THAT MUST NOT HAVE OCCURRED
1	Start launch received		2
2	Battery activate	1	4
3	Missile enable	1-2	4
4	First motion	1-3	
5	Launch escape	1-4	
6	Booster acceleration	1-5	
7	Launcher clear	1-6	
8	Booster thrust decay	1-7	
9	Transition complete	1-8	
10	Maneuvering speed	11	
11	Arming maneuver position	1-9	
12	Arming maneuver	14	
13	Left turn	10, 11	
14	Right turn	13	
15	Final arm position	1-9, 16-23	
16-22	Events associated with terrain correlation and voting	1-9	
23	Lateral steering	1-9, 16-22	
24	In-flight BIT	1-9, 15-23	

Figure 29. Example of ESD Word Formation Requirements

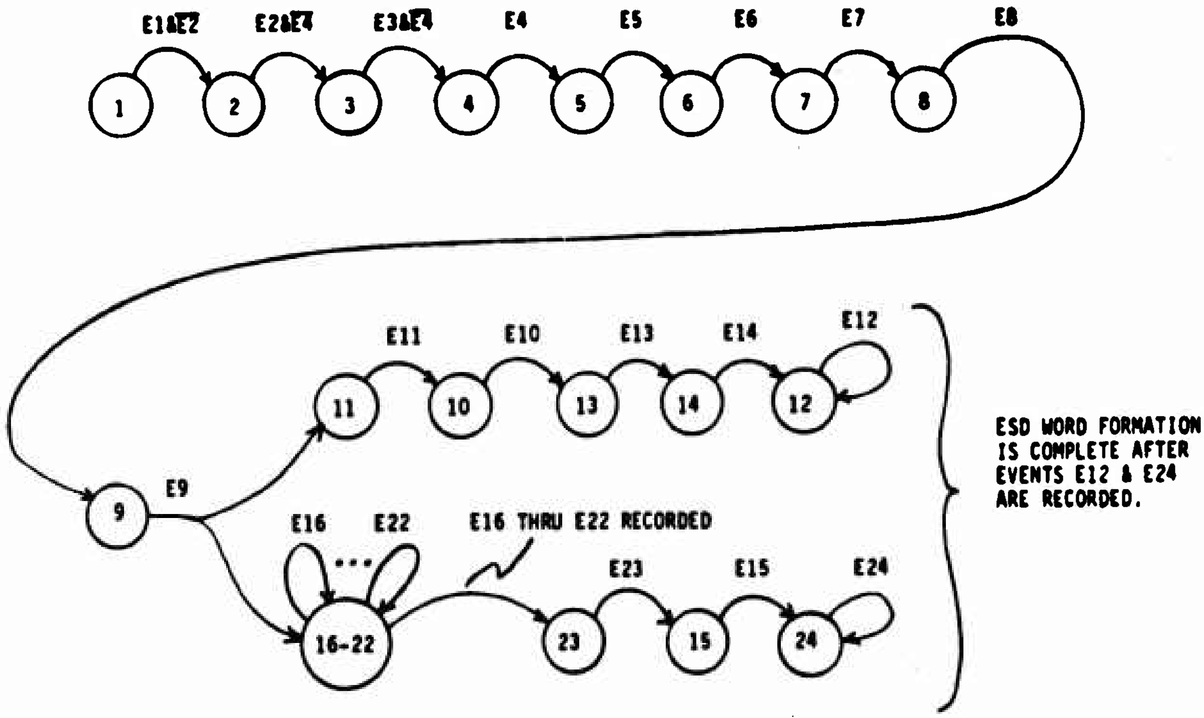

Figure 30. Mealy Machine that Forms TLAM-N/G ESD Word

The Target Countdown Sequencing function consists of a calculation of time-to-go to the target and a timed sequence of discrete outputs which detonate the warhead. As for the above case, this function is identical for the two missiles and a simple part could be specified to serve the needs of both missiles.

a. Application of Nuclear Parts to Other Missiles

CAMP nuclear warhead control parts are applicable to new missiles outside the CAMP missile set if the new missiles use current generation nuclear warheads such as the W80 warhead and the W84 warhead. In the case of the ESD Word Formation function, the unique flight events associated with a new missile would likely be different than those for TLAM-G and TLAM-N. However, the CAMP Mealy machine schematic part could still be used to construct a part to satisfy the ESD word formation requirements of the new missile.

For the remaining four nuclear warhead functions (Warhead Arming Maneuver, Function Select Sequencing, Final Warhead Arming Sequencing, and Target Count Down Sequencing), the reusability of the CAMP parts depends on the degree of similarity of the new missile's warhead with the W80 and W84 warhead. In one extreme, if the new missile happened to use the W80 or W84 warhead directly, then the CAMP abstract process parts for event sequencing would still be useful in constructing these four warhead control functions, since they have (for the most part) a sequential structure.

At the other extreme, consider a new missile which uses a "next" generation nuclear warhead. SRAM-II (Short Range Attack Missile) is in this category. The next generation nuclear warhead, as envisioned at the present time for SRAM-II, has its own embedded computer which communicates with the main missile computer via a data bus. While details of the warhead control functions performed by the warhead computer are sketchy at this time, it is likely that the CAMP abstract process parts (e.g., the schematic Mealy machine part) could be used to help construct the software resident in the nuclear warhead computer.

Except for the Target Count Down Sequencing function, the nature of the warhead support software resident in the main missile computer will change. Once this support software is clearly defined, it is probable that reusable parts could be designed to help implement the warhead support function in missiles with next generation nuclear warheads.

3. CONVENTIONAL/PROGRAMMED DETONATION

TLAM-C is the only missile in the CAMP set whose warhead operation falls in the category of conventional/programmed detonation. In this case a conventional warhead is used, without the aid of a terminal seeker, against a fixed target. The goal of the warhead control software is to denonate the warhead so that the resulting fragments impact the target as directly as possible. The following operations have been identified to implement the warhead control function for missiles in this category:

-- Arming Sequencing

-- Detonation Time Calculations

  -- Compute Dynamic Warhead Fragment Angle

  - Compute Warhead Burst Altitude

  -- Compute Detonation Time

-- Detonation Sequencing

At the top level these functions can be structured as the state machine shown in Figure 31. Again the CAMP Mealy machine part could be used for this purpose.

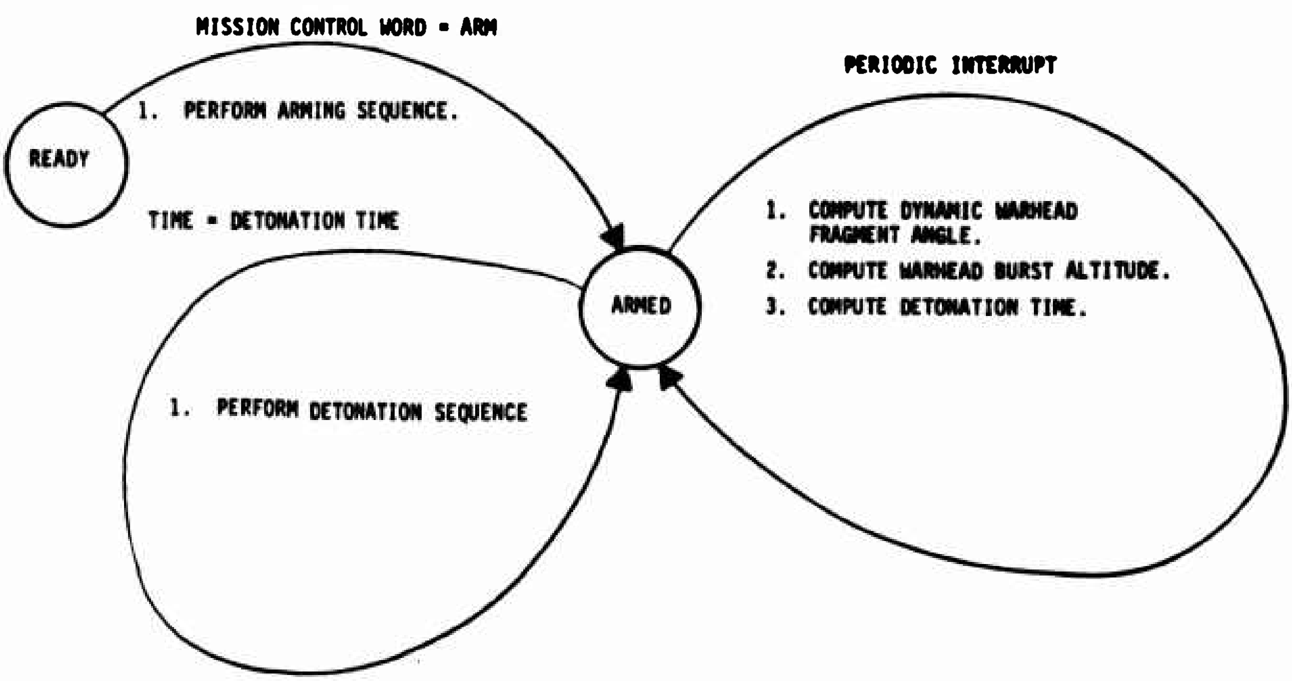

Figure 31. Conventional/Programmed Detonation State Diagram

The Arming Sequencing function issues a timed sequence of outputs to arm the warhead, while the Detonation Sequencing function issues a timed sequence of outputs to detonate the warhead. Each function could be constructed using the CAMP abstract process parts for event sequences and time driven event sequences.

The three computations listed under Detonation Time Calculations would be constructed as three separate simple parts which perform the following functions:

-- Compute Dynamic Warhead Fragment Angle considers the mean warhead fragment velocity, the missile horizontal and vertical velocities, missile pitch attitude, and the warhead bias angle to compute the angle (with respect to the vertical) of the trajectory of warhead fragments.
-- Compute Warhead Burst Altitude computes the altitude with respect to the target of the point of warhead detonation.
-- Compute Detonation Time considers geometric parameters of the target engagement to compute the time of warhead detonation that will provide the most direct impact of the target by warhead fragments.

a. Application of Conventional/Programmed Detonation Parts to Other Missiles

The use of CAMP Conventional/Programmed Detonation warhead control parts is applicable to missiles outside the CAMP missile set which employ warheads in this category. For the Arming Sequencing and Detonation Sequencing functions, the CAMP abstract process parts for event sequences and time driven sequences could be used to meet the requirements of a new missile's warhead control. It is likely that the simple parts under Detonation Time Calculations could be used directly.

4. CONVENTIONAL/IMPACT DETONATION

MRASM-L, TASM, and Harpoon are the missiles whose warhead control is in this category. Here, the warhead is armed by a simple sequence of steps

after the occurrence of a certain event, such as seeker lock-on. This Arming Sequence function, as already discussed above, would use the CAMP abstract process parts.

a. Application of Conventional/Impact Detonation Parts to Other Missiles

Because of the simplicity of the warhead control functions in this category, the CAMP part usage discussed above is, most likely, also valid for missiles outside the CAMP missile set.

5. SUBMUNITION

MRASM-L is the only missile in the CAMP missile set whose warhead operation falls in the category of submunition. Analysis of this area is currently incomplete, however the following functions have been identified:

-- Compute Time-at-Target Centroid
-- Issue Warhead Dispenser Arming Command
-- Set Over-Runway-Flag
-- Estimate Wind at Target

Each of these functions could be implemented by a simple part.

a. Application of Submunition Parts to Other Missiles

Although analysis is incomplete, it is reasonable to expect that most of the simple parts identified in the submunition warhead category would be useful for new missiles which employ submunition warheads.

SECTION IX
AIR DATA PARTS RATIONALE

1. INTRODUCTION

Air data software provides aerodynamic information such as mach number and dynamic pressure for use by other missile software functions. The inputs to the air data software consist of voltages representing measurements made by the air data instruments of impact pressure, static pressure, and temperature. The air data software processes this information to provide mach number, dynamic pressure, barometric altitude, and the speed of sound.

The six software parts shown in Figure 32 have been identified to help the software engineer implement the air data function.

Part R228 - Compute Outside Air Temperature
Part R229 - Compute Pressure Ratio
Part R230 - Compute Mach
Part R231 - Compute Dynamic Pressure
Part R232 - Compute Speed of Sound
Part R233 - Compute Barometric Altitude

Figure 32. Air Data Parts

2. PARTS OPERATION

The top level operation of the air data parts is shown in Figure 33. Part R106 (a math part called External Form Conversion) rescales the voltages received from the air data instruments to produce Impact_Pressure, Measured Static_Pressure, and Total_Temperature.

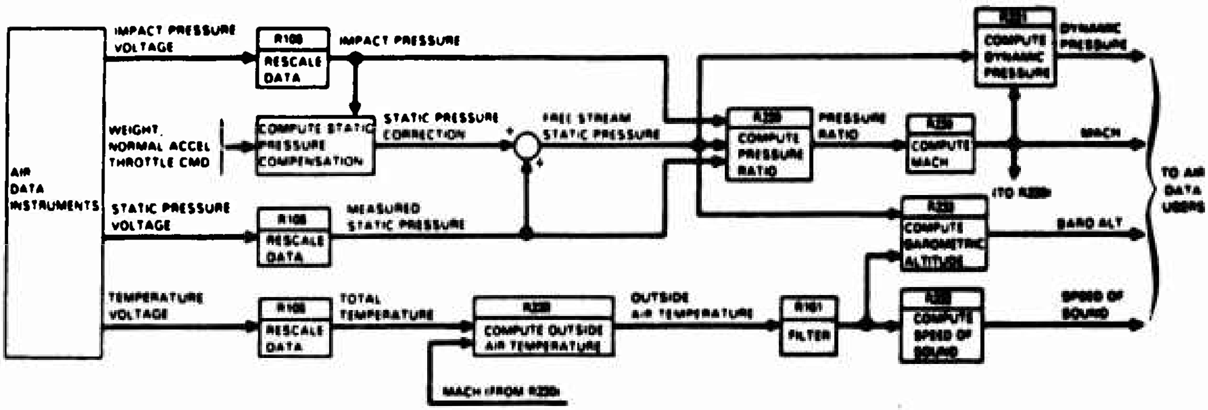

Figure 33. Air Data Parts Operation

A missile peculiar software function computes a correction to Measured Static_Pressure which removes disturbances to the air stream caused by the missile airframe. After this correction is applied, Measured_Static_Pressure becomes Free_Stream_Static_Pressure. This quantity, along with Impact Pressure and Measured_Static_Pressure is used by part R229 to compute Pressure_Ratio. In turn, Pressure_Ratio is the input to Part R230 which produces Mach.

Using Mach and Free_Stream_Static_Pressure, Part R231 Computes Dynamic Pressure. Mach is also used by Part R228 to compute Outside_Air_Temperature from Total_Temperature. The Outside_Air_Temperature is then filtered by part R161 (which is a signal processing part called Tustin Lag Filter).

The filtered Outside_Air_Temperature is used by part R232 to compute Speed_of_Sound. It is also used, by part R229, in combination with Free Stream_Static_Pressure to produce Baro_Alt (Barometric Altitude).

3. PARTS DESCRIPTION

a. Compute Outside Air Temperature (Part R228)

This part computes the temperature of the air at a location which is far enough away from the missile so that the air temperature is unaffected by the missile airframe. While this air temperature is called outside air tempera- ture, it is also known as free stream air temperature. In performing this computation, the part uses total temperature, mach number, and recovery fac- tor. A typical value for recovery factor is 0.93 but the exact value varies slightly from one missile to the next depending on the particular air data instrument installation.

b. Compute Pressure Ratio (Part R229)

This part computes the ratio of total pressure to free stream static pressure given measured static pressure, impact pressure, and free stream static pressure. Note that total pressure is the sum of measured static pressure and impact pressure

c. Compute Mach (Part R230)

This part computes mach from pressure ratio by using an approximation to the following exact expression:

$$\text{Mach} = \text{Sqrt}\ (\ 5 * [\ (\ \text{Pressure\_Ratio}\ )^{2/7} - 1\ ]\ )$$

The three constants (C_0, C_1, and C_2) used in the approximation are selected to provide the best approximation over the mach range of interest. For example, with $C_0 = -1.65722$, $C_1 = 1.94666$, and $C_2 =$ -.287428 the approximation used by this part is within 0.26 percent of the exact expression over a mach range from 0.6 to 0.95.

Note also that a multiplicative factor to compensate pressure ratio can be embedded in the coefficients C_1 and C_2 if required.

d. Compute Dynamic Pressure (Part R231)

This part computes dynamic pressure given free stream static pressure and mach number.

e. Compute Speed of Sound (Part R232)

This part computes the speed of sound in air from the temperature of the air.

f. Compute Barometric Altitude (Part R233)

This part computes barometric altitude by numerical integration of the atmospheric equation of state based on free stream static pressure and outside air temperature (also called free stream air temperature).

4. PARTS USAGE

The application, with respect to the CAMP missile set, of the air data parts discussed above is shown in Figure 34. Generally the air data parts provided are useful for any missile that employs an air data system to give any or all of the following information: dynamic pressure, mach number, barometric altitude, and speed of sound.

A: UTG SINP B: AGM-109H C: AGM-109L D: BGM-109C E: BGM-109B

F: BGM-109G G: HARPOON H: BGM-109A I: MGD GANP J: SPARTAN

Part	Description	A	B	C	D	E	F	G	H	I	J
Part R228	Compute Outside Air Temperature........		X	X	X		X		X		
Part R229	Compute Pressure Ratio.		X	X	X	X	X		X		
Part R230	Compute Mach...........		X	X	X	X	X		X		
Part R231	Compute Dynamic Pressure...............		X	X	X	X	X		X		
Part R232	Compute Speed of Sound.		X	X	X		X		X		
Part R233	Compute Barometric Altitude...............		X	X	X		X		X		

Figure 34. Air Data Parts Usage

SECTION X

FUEL CONTROL PART RATIONALE

1. INTRODUCTION

Part R234 (Compute Throttle Command to Control Mach) implements a mach control loop whose purpose is to control missile mach number to a desired value. To do this, the part controls fuel flow to the engine by adjusting the engine throttle position.

The desired mach number (mach command) is chosen external to this part to optimize fuel usage or to control waypoint arrival times.

2. PART DESCRIPTION

a. Compute Throttle Command to Control Mach (Part R234)

As shown in Figure 35 the mach control loop implemented by this part computes the error between mach command and mach feedback. The lower level part R048 (Integral Plus Proportional Gain) is used to drive the mach error to zero by commanding throttle movements in the appropriate direction. After the throttle command (labeled Raw_Throttle_Cmd in Figure 35) is generated, it is processed by a first order filter which serves three purposes:

-- The filter limits the maximum rate of change of the throttle command,

-- It limits the bandwidth of the throttle command signal, and

-- It limits the maximum and the minimum throttle command to the physical limits of throttle movement.

Other significant features of this part are:

-- The throttle integrator (in the lower portion of Figure 35) is frozen when its output, the throttle command, is in the limited condition.

-- The mach error integrator (in the upper portion of Figure 35) is frozen when either the throttle command is in the limited condition, or when the mach error integrator output is in the limited condition.

-- The user is provided with an initialization operation which sets the initial value for each of the two integrators so that there is no abrupt change in the throttle command when the mach control loop is first activated.

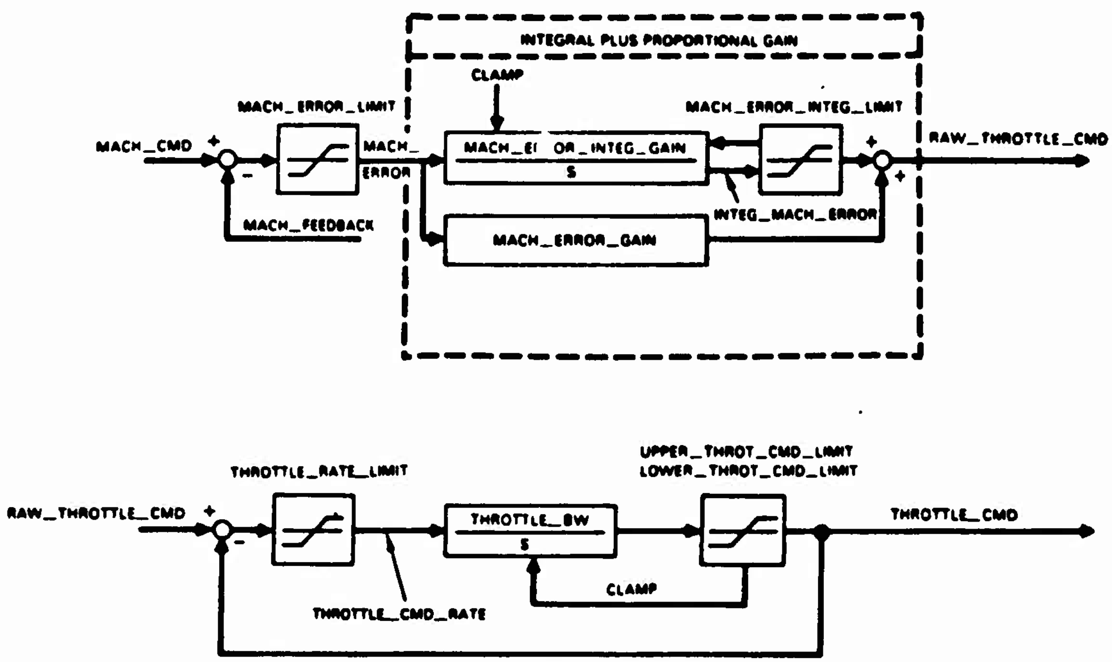

Figure 35. Mach Control Loop

3. PART USAGE

The application with respect to the CAMP missile set of the part discussed above is shown in Figure 36. Generally Part R234 is a candidate for use in any missile which has a throttable engine and which requires the control of mach number.

A: UTG SINP B: AGM-109H C: AGM-109L D: BGM-109C E: BGM-109B
F: BGM-109G G: HARPOON H: BGM-109A I: MGD GANP J: SPARTAN

	A	B	C	D	E	F	G	H	I	J
R234 Compute Throttle Command to Control Mach............		X	X	X	X	X		X		

Figure 36. Fuel Control Part Usage

SECTION XI
ABSTRACT DATA STRUCTURE PARTS RATIONALE

1. INTRODUCTION

Abstract Data Structures provide a data structure and operations to be performed on that data. Outside components may only access the abstract data structure through subprograms which form the operations of the abstract structure. These subprograms should offer primitive operations on the structure from which an outside program can build higher level operations to meet its own processing needs.

The advantage to defining abstract data structures is that they provide a means of organizing a complex program using a standard set of operations. Their ability to limit the scope of interfaces to a given set of data simplifies the design of user programs and guarantees independence from the underlying data structure and algorithms. The potential user of these parts must be aware of such features as memory requirements, algoritmic complexity, and data security. With these facts the user can choose the appropriate data structure and implementation for his program, and should his performance or operational needs change, he can change Abstract Data Structures with minimal on his application program.

A message buffering example will explain the nature of these parts. If the user knows he has four different message streams, then he must declare four message buffers to handle this data. If the user knows the maximum number of messages which he will ever want to buffer, then he will likely choose a buffer with static size. If, however, the number of messages in the buffer fluctuates across a wide range, then allocating maximum size buffers may be costly in terms of memory. If the user later decides that two of the buffers should be unbounded, because they can grow and shrink dynamically, then he may choose to change the data type for those two buffers. The operations on the unbounded buffer will be essentially the same as those on

the bounded, except for test for full. Now the user has changed his architectural design, but with minimal impact on the rest of his system.

Figure 37 shows the parts which make up the data structure parts. The paragraph numbers refer to section numbers in the CAMP Software Requirements Specification.

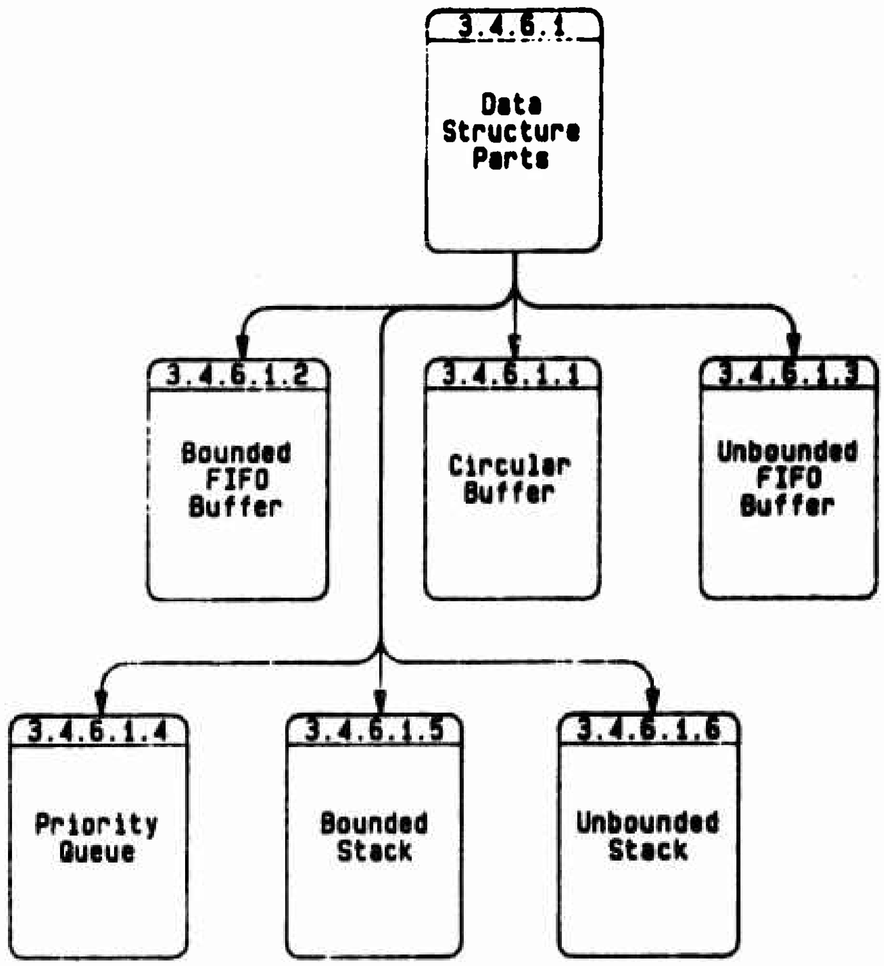

Figure 37. Organization of Data Structure Parts

2. PARTS DESCRIPTIONS

The Abstract Data Structure Parts consist of Buffers, Queues and Stacks used for missile control functions. The domain analysis revealed the parts shown in Figure 38.

A: UTG SINP B: AGM-109H C: AGM-109L D: BGM-109C E: BGM-109B

F: BGM-109G G: HARPOON H: BGM-109A I: MGD GANP J: SPARTAN

	A	B	C	D	E	F	G	H	I	J
Bounded FIFO Buffer	X			X	X	X		X		
Circular Buffer									X	
Unbounded FIFO Buffer										
Priority Queue		X	X	X	X	X		X		
Bounded Stack					X	X		X		
Unbounded Stack										

Figure 38. Data Structure Parts Utilization

a. Bounded FIFO Buffer

The Bounded FIFO Buffer will perform buffering of data in a first in first out fashion. The part will limit the number of items which may be in the buffer at any one time and will raise an exception if an attempt is made to add to an already full buffer. The part can be used to buffer incoming Mission Data, TERCOM processing or DSMAC updates. In addition, this part can be used for message passing between components of a system.

b. Circular Buffer

The circular buffer is a particular version of a bounded FIFO buffer. The chief difference is that the circular buffer will not block incoming data if the buffer is already full. Rather, it will overwrite older data to assure that all new data will be seen. This part may be used to buffer the same types of data as the Bounded FIFO buffer.

c. Unbounded FIFO Buffer

This part performs the same functions as the Bounded FIFO Buffer. It differs from the bounded version in that it can accept an unlimited number of inputs to the buffer. The Unbounded FIFO Buffer will be used in future missiles to control mission data created after the missile is in flight.

d. Priority Queue

The Priority Queue performs buffering information on incoming data streams. It differs from the FIFO buffers in that incoming data carries a priority. The priority will determine the order in which the queue will process data. For the current missiles the priority queues are used by the runtime executive to schedule tasks and controlling data flow. Future missiles will use this structure to control external message handling and to support dynamic task priorities in Ada.

e. Bounded Stack

This part performs stack operations on incoming data. The bounded stack will raise an overflow condition if an attempt is made to push data on an already full stack. It will raise an underflow condition if an attempt is made to remove data from an empty stack. The missiles in the CAMP domain could use a stack data structure to control switch settings for event control. In future missiles they could be used to perform command processing after launch.

f. Unbounded Stack

This part performs the same operations as the Bounded Stack. The size of the stack is, however, dynamic. It will grow to accept additional incoming data as required. The Unbounded Stack will raise only the underflow condition if an attempt is made to remove data from an empty stack.

SECTION XII
MATHEMATICAL PARTS RATIONALE

1. INTRODUCTION

The general purpose mathematics package has, in the past, been the one area of software development which has promoted the highest level of reuse. For mainframe system applications, this package has consisted of a wide range of off-the-shelf routines, thoroughly tested to provide accurate results with minimum overhead. For embedded systems, this package has provided a more restricted range of reusable algorithmic designs, to guarantee results which are sufficiently accurate to the demands of the system.

The CAMP math parts provide off-the-shelf routines, coded in Ada, to satisfy the computational requirements of our missile domain. Many of these parts are those typically found in general purpose math packages. The trigonometric functions, for example, will be implemented though lower-level polynomial or system functions to provide a standard set of well-tested parts guaranteeing a high degree of accuracy with minimum overhead. In addition, the design will supply the polynomial functions themselves to the part user so that he might custom tailor his own special-purpose algorithms. Other parts, such as signal processing and data conversion, are less commonly found in general packages, but are essential for missile applications.

This rationale will present each of the major part groups and explain their significance to missile applications. With the exception of signal processing parts, some or all of the parts are used in each of the ten missiles. In many cases, the CAMP study has included parts which are not used in any single missile but are certainly appropriate for the applications. Parts have been added to create a complete set of parts for each group. (For example, many of the unit conversion parts are extrapolated for conversions which may be needed in the future.) For signal processing parts a table of parts usage will show which missiles use these parts.

2. ISSUES

a. Requirements for the CAMP Domain

The CAMP domain analysis has shown that there are a wide range of mathematical parts which the missile domain will require. The organization of these parts, as documented in the Software Requirements Specification is shown in Figure 39. Paragraph 3 explains the significance of each of these parts and how they will be used to support other CAMP parts.

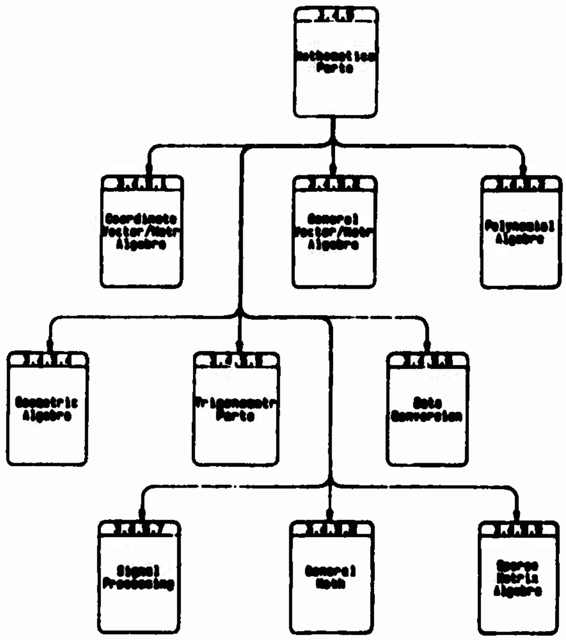

Figure 39. Organization of CAMP Mathematical Parts

b. Accuracy, Precision, and Overhead of Numerical Methods

A major set of issues which the CAMP parts for mathematical operations must address relate to the accuracy, precision and overhead incurred by these operations. Because these parts are often used in time-critical computations, there must be minimal cost in using the parts. This issue has been discussed in Volume I of this report and below under those parts which are specifically designed to meet timing constraints.

In addition to time criticality, there is also a need for parts which can address precision and accuracy of computation. The factors influencing this area are the data types for objects (their precision) and the operations which may be performed on those objects. The CAMP structure supports special purpose polynomial functions through the Polynomial Algebra parts which afford the user great flexibility in defining his own set of operations to meet the demands for accurate computation. The rationale for the polynomial parts is also discussed below.

c. **Reuse of CAMP Math Parts within the CAMP Domain**

A final area of concern are those parts which specifically satisfy the computational needs of other parts in the CAMP domain. Figure 40 shows the mathematics parts which support one or more other CAMP parts. The reused math parts have been required to meet the particular needs of other parts in the CAMP domain. The method of design allows for integration with other CAMP parts and there is a strong dependence upon the particular design structure as described in the Top-Level Design Document. But it is significant that the math parts will easily lend themselves to supporting operations from outside the CAMP domain as long as the non-CAMP parts are designed according to the CAMP methods.

CAMP MATH PART	SUPPORTED PARTS
Coordinate Vector/ Matrix Algebra	Navigation, Waypoint Steering
General Vector/ Matrix Algebra	Kalman Filter
Polynomial Algebra	Trigonometric Parts
Geometric Albegra	Waypoint Steering
Signal Processing	Autopilot, Mach Control
Sparse Matrix Algebra	Kalman Filter

Figure 40. CAMP Parts Supported by the Mathematical Parts

3. PARTS DESCRIPTION

a. Coordinate Vector/Matrix Algebra Parts

These parts support four types of operations: Vector-Scalar, Vector-Vector, Vector-Matrix, and Matrix-Matrix. The guiding principle of these parts has been the need for efficiency in computation. This particular set of parts takes advantage of the fact that most Vector/Matrix operations performed in missile systems involve a three dimension coordinate frame. The parts can then unfold operations on the Vector or Matrix to avoid looping tests. The time advantage gained is to reduce an n^2-complexity operation to an n-complexity operation. These concepts are fully discussed in Section IV of Volume I of this report.

b. General Vector/Matrix Algebra Parts

These parts perform the same set of operations as those of the Coordinate Vector/Matrix Algebra Parts. They will be used for applications which need general, rather than coordinate, operations. In addition, these parts will define special purpose operations for Diagonal and Symmetric matrices.

c. Polynomial Algebra

A series of polynomial solutions underlies the trigonometric and other transcendental operations. These methods have been developed by numerical analysts to provide solutions which can alternatively offer high accuracy, computation speed, or memory efficiency. A system-defined trigonometric operation will use one such method to offer a compromise of these criteria.

The rationale behind the Polynomial parts which are shown in Figure 41 is to offer a variety of approaches to solve trigonometric and other similar problems. Where the trigonmetric parts offer a single solution, the

polynomial parts offer a range of different solutions depending on the particular method chosen by the user. The user may need a sine which is extremely accurate, and he is willing to pay the time penalty for this computation. The Polynomial parts will offer him such a function or he can use the parts to define his own.

Table Lookup	Least Squares
Chebyshev	Legendre
Fike	Modified Newton-Raphson
Hart	Newton-Raphson
Hastings	Taylor Series
System Functions	

Figure 41. Polynomial Parts

A special component within these parts is the Table Lookup functions. This component offers standard tables for use in finding solutions to trigonometric problems. The par. also gives the user the ability to create his own table of values or to pass in an existing table. This will afford the user great flexibility in defining table-based operations.

d. Geometric Algebra

This area exists primarily to support the Waypoint steering operations which must perform computations based on earth geometry. The particular demands of Waypoint steering lead to the creation of highly specialized geometric operations, however, and for the most part these are included under the Waypoint Steering operations.

e. Trigonometric Parts

These parts support all of the trigonmetric requirements of the missile domain. The parts are not a comprehensive collection of trigonmetric

operations, however. The CAMP parts are limited to those needed to support missile-related operations. These are shown in Figure 42.

The Sine-Cosine and Arcsine-Arccosine are another case of designing parts for efficiency. These parts exist to take advantage of the fact that intermediate values in the computation of sine and cosine, and in their inverses, are identical. Computing both values at once can save numerous multiplies and adds.

Sine	Arcsine
Cosine	Arccosine
Tangent	Arctangent
Sine-Cosine	Arcsine-Arccosine

Figure 42. Trigonometric Parts

f. Data Conversion

There are two types of parts which fall under this heading. The first deals with parts which are needed to perform unit conversion, e.g. feet to meters. These conversions are very simple and rely on conversion constants declared in other parts.

The second type of part performs conversion on external interface data. This data comes into a missile system over analog-to-digital channels and must be converted from engineering units to internal form, generally floating point. Data being passed as output from the system to external devices must be converted from the internal form to the external form. The data conversion parts are required to accept parameters which will permit easy conversion in both directions on external data.

g. Signal Processing

The Signal Processing parts consist of Limiters and Filters which are used in signal processing operations for missile control functions. The domain analysis revealed the parts shown in Figure 43.

A: UTG SINP	B: AGM-109H	C: AGM-109L	D: BGM-109C	E: BGM-109B
F: BGM-109G	G: HARPOON	H: BGM-109A	I: MGD GANP	J: SPARTAN

	A	B	C	D	E	F	G	H	I	J
Limiter (Upper and Lower Bounds)		X	X	X	X	X	X	X		
Limiter (Upper Bound)		X	X	X	X	X		X		
Limiter (Lower Bound)										
Absolute Limiter		X	X	X	X	X		X		
Absolute Limiter with Flag		X	X	X	X	X		X		
General First Order Filter		X	X							
Tustin Lag Filter		X	X	X	X	X		X		
Tustin Lead Lag Filter		X	X	X	X	X		X		
Second Order (Notch) Filter				X		X		X		
Tustin Integrator with Limit				X		X		X		

Figure 43. Signal Processing Parts Usage

(1) Limiter (Upper and Lower Bounds)

This part will provide a limiter operation giving the user the ability to specify both upper and lower bounds.

(2) Limiter (Upper Bound)

This part will provide a limiter operation giving the user the ability to specify an upper bound only.

(3) Limiter (Upper Bound)

This part will provide a limiter operation giving the user the ability to specify a lower bound only.

(4) Absolute Limiter

This part will provide a limiter operation giving the user the ability to specify a single bound. This bound will be compared to the absolute value of the incoming signal to perform the limiter operation.

(5) Absolute Limiter with Flag

This part will provide a limiter operation giving the user the ability to specify a single bound. This bound will be compared to the absolute value of the incoming signal to perform the limiter operation. If the bound is reached in either direction, a flag is set to indicate the limit condition. The bound is cleared once the value falls below the limit.

(6) General First Order Filter

This part will perform a first order filter operation.

(7) Tustin Lag Filter

This part will perform a first order filter operation using the Tustin Lag method.

(8) Tustin Lead Lag Filter

This part will perform a first order filter operation using the Tustin Lead Lag method.

(9) Second Order (Notch) Filter

This part will perform a second order filter operation using the Tustin Lead Lag method.

(10) Tustin Integrator with Limit

This part will perform an integration operation and perform an absolute limit operation as defined in paragraph (5).

h. General Purpose Math

These parts include a variety of general purpose math operations as shown in Figure 44.

i. Static Sparse Matrix Algebra

This part supports the Kalman Filter operations. It is a schematic part which unfolds a sparse matrix into a record of scalar elements. The part also includes operations on a static sparse matrix. The role of this part in the CAMP parts is fully explained in Appendix II of the Software Requirements Specification of the Software Parts Constructor.

Interpolate	Change Accumulator
Extrapolate	Square Root
Decrement	Absolute Value
Running Average	Root Sum of Squares
Accumulator	Integrator

Figure 44. General Purpose Math Parts

THE APPENDIX

CAMP PARTS LIST

This appendix contains a list of parts identified during the CAMP project. The parts are listed by category and subcategory. The part number is a unique number used for identification purposes. The column entitled Schematic Only refers to whether a part consists only of a part constructor or whether there is also an underlying part that is available separately from the constructor.

CATEGORY	SUBCATEGORY	PART #	NAME	Schematic Only
Data	Constants	R017	WGS72 Ellipsoid Data (Metric Version)	No
Data	Constants	R018	WGS72 Ellipsoid Data (Engineering Version)	No
Data	Constants	R019	WGS72 Unitless Ellipsoid Data	No
Data	Constants	R020	Universal Constants	No
Data	Constants	R158	Conversion Factors	No
Data	Types	R021	Basic Data Types	No
Data	Types	R023	Kalman Filter Data Types	No
Data	Types	R163	Autopilot Data Types	No
Data	Types	R236	Data Type Constructor	Yes
Equipment Interface	General Purpose	R189	Missile Radar Altimeter Handler	No
Equipment Interface	General Purpose	R042	Missile Radar Altimeter Handler with Auto Power On	No
Equipment Interface	General Purpose	R043	Bus Interface Constructor	Yes
Equipment Interface	General Purpose	R046	Clock Handler	No
Navigation	Wander Azimuth	R185	Compute East Velocity	No
Navigation	Wander Azimuth	R186	Compute North Velocity	No
Navigation	Wander Azimuth	R001	Compute Earth Relative Horizontal Velocities	No
Navigation	Common	R002	Altitude Integration	No
Navigation	Common	R003	Compute Ground Velocity	No
Navigation	Wander Azimuth	R004	Compute Total Angular Velocity	No
Navigation	Common	R005	Compute Gravitational Acceleration	No
Navigation	Common	R006	Compute Gravitational Acceleration from Sin LAT	No
Navigation	Wander Azimuth	R007	Compute Coriolis Acceleration	No
Navigation	Wander Azimuth	R187	Compute Coriolis Acceleration from Total Rates	No
Navigation	North Pointing	R008	Compute Coriolis Acceleration	No
Navigation	Common	R009	Compute Heading	No
Navigation	Common	R010	Update Velocity	No
Navigation	Common	R039	Compute Scalar Velocity	No
Navigation	Wander Azimuth	R035	Compute Radii of Curvature	No
Navigation	North Pointing	R036	Compute Radii of Curvature	No
Navigation	North Pointing	R012	Compute Total Platform Rotation Rates	No
Navigation	Wander Azimuth	R011	Compute Total Platform Rotation Rates	No
Navigation	Common	R157	Compute Rotational Increments	No
Navigation	North Pointing	R014	Compute Earth Rotation Rate	No
Navigation	Wander Azimuth	R013	Compute Earth Rotation Rate	No
Navigation	Wander Azimuth	R025	Compute Earth Relative Rotation Rates	No
Navigation	North Pointing	R026	Compute Earth Relative Rotation Rates	No
Navigation	North Pointing	R027	Latitude Integration	No

CATEGORY	SUBCATEGORY	PART #	NAME	Schematic Only
Navigation	Wander Azimuth	R029	Compute Latitude	No
Navigation	Wander Azimuth	R030	Compute Latitude using Arctangent	No
Navigation	North Pointing	R031	Longitude Integration	No
Navigation	Wander Azimuth	R033	Compute Longitude	No
Navigation	Wander Azimuth	R028	Compute Wander Azimuth Angle	No
Navigation	Common	R032	Initialize Direction Cosine Matrix	No
Navigation	Common	R034	Trapeziodal Integration	No
Navigation	North Pointing	R016	North Pointing Navigation Bundle	No
Navigation	Common	R188	Common Navigation Bundle	No
Navigation	Wander Azimuth	R015	Wander Angle Navigation Bundle	No
Navigation	Common	R237	Navigation Component Constructor	Yes
Navigation	Common	R238	Navigation Subsystem Constructor	Yes
Kalman Filter	General	R145	Propogate State Transition and Process Noise Matrices	No
Kalman Filter	General	R146	Propogate Error Covariance Matrix	No
Kalman Filter	General	R147	Kalman Update	No
Kalman Filter	General	R148	Propogate State Transition Matrix	No
Kalman Filter	General	R149	Compute Kalman Gains	No
Kalman Filter	General	R150	Update Error Covariance Matrix	No
Kalman Filter	General	R151	Update State Vector	No
Kalman Filter	General	R152	Sequentially Update Covariance Matrix and State Vector	No
Kalman Filter	General	R153	Kalman Executive	No
Kalman Filter	Complicated H	R181	Kalman Update	No
Kalman Filter	Complicated H	R182	Compute Kalman Gain	No
Kalman Filter	Complicated H	R183	Update Error Covariance Matrix	No
Kalman Filter	Complicated H	R184	Update State Vector	No
Kalman Filter	Complicated H	R201	Sequentially Update Covariance Matrix and State Vector	No
Math	Coordinate Algebra	R047	Coordinate Algebra Bundle	No
Math	Coordinate Algebra	R050	Vector Vector Addition	No
Math	Coordinate Algebra	R205	Sparse Right X Vector-Vector Addition	No
Math	Coordinate Algebra	R206	Sparse Right Z Vector-Vector Addition	No
Math	Coordinate Algebra	R051	Vector Vector Subtraction	No
Math	Coordinate Algebra	R207	Sparse Right XY Vector-Vector Subtraction	No
Math	Coordinate Algebra	R052	Vector Vector Dot Product	No
Math	Coordinate Algebra	R208	Vector Length	No
Math	Coordinate Algebra	R053	Vector Vector Cross Product	No
Math	Coordinate Algebra	R054	Vector Scalar Multiplication	No
Math	Coordinate Algebra	R209	Sparse X Vector-Scalar Multiplication	No

CATEGORY	SUBCATEGORY	PART #	NAME	Schematic Only
Math	Coordinate Algebra	R055	Vector Scalar Division	No
Math	Coordinate Algebra	R049	Matrix Vector Multiplication	No
Math	Coordinate Algebra	R056	Matrix Scalar Multiplication	No
Math	Coordinate Algebra	R057	Matrix Scalar Division	No
Math	Coordinate Algebra	R060	Matrix Scalar Addition	No
Math	Coordinate Algebra	R067	Matrix Scalar Subtraction	No
Math	Coordinate Algebra	R068	Matrix Matrix Multiplication	No
Math	Coordinate Algebra	R070	Matrix Matrix Addition	No
Math	Coordinate Algebra	R071	Matrix Matrix Subtraction	No
Math	Coordinate Algebra	R072	Set to Identity Matrix	No
Math	Coordinate Algebra	R078	Set to Zero Matrix	No
Math	Matrix Algebra	R058	Matrix Algebra Package	No
Math	Matrix Algebra	R061	Vector Vector Addition	No
Math	Matrix Algebra	R062	Vector Vector Subtraction	No
Math	Matrix Algebra	R063	Vector Vector Dot Product	No
Math	Matrix Algebra	R104	Vector Length	No
Math	Matrix Algebra	R065	Vector Scalar Multiplication	No
Math	Matrix Algebra	R066	Vector Scalar Division	No
Math	Matrix Algebra	R069	Matrix Vector Multiplication	No
Math	Matrix Algebra	R073	Matrix Scalar Multiplication	No
Math	Matrix Algebra	R074	Matrix Scalar Division	No
Math	Matrix Algebra	R075	Matrix Scalar Addition	No
Math	Matrix Algebra	R076	Matrix Scalar Subtraction	No
Math	Matrix Algebra	R077	Matrix Matrix Multiplication	No
Math	Matrix Algebra	R079	Matrix Matrix Addition	No
Math	Matrix Algebra	R080	Matrix Matrix Subtraction	No
Math	Matrix Algebra	R155	Set to Identity Matrix	No
Math	Matrix Algebra	R156	Set to Zero Matrix	No
Math	Matrix Algebra	R210	Statically Sparse Matrix Constructor	Yes
Math	Matrix Algebra	R226	Define Dynamically Sparse Matrix	No
Math	Matrix Algebra	R211	Define Symmetric (Half Storage) Matrix	No
Math	Matrix Algebra	R227	Define Symmetric (Full Storage) Matrix	No
Math	Matrix Algebra	R212	Define Diagonal Matrix	No
Math	Matrix Algebra	R235	General Purpose Matrix Constructor	Yes
Math	Geometric	R081	Geometric Package	No
Math	Geometric	R082	Compute Angle Between Headings	No
Math	Trigonometric	R083	Radian Trigonometric Package	No

CATEGORY	SUBCATEGORY	PART #	NAME	Schematic Only
Math	Trigonometric	R084	Degree Trigonometric Package	No
Math	Trigonometric	R085	Semicircle Trigonometric Package	No
Math	Trigonometric	R086	Sine	No
Math	Trigonometric	R087	Cosine	No
Math	Trigonometric	R088	Tangent	No
Math	Trigonometric	R089	Arcsine	No
Math	Trigonometric	R090	Arccosine	No
Math	Trigonometric	R091	Arctangent	No
Math	Trigonometric	R092	Sine	No
Math	Trigonometric	R093	Cosine	No
Math	Trigonometric	R094	Tangent	No
Math	Trigonometric	R095	Arcsine	No
Math	Trigonometric	R096	Arccosine	No
Math	Trigonometric	R097	Arctangent	No
Math	Trigonometric	R098	Sine	No
Math	Trigonometric	R099	Cosine	No
Math	Trigonometric	R100	Tangent	No
Math	Trigonometric	R101	Arcsine	No
Math	Trigonometric	R102	Arccosine	No
Math	Trigonometric	R103	Arctangent	No
Math	Data Conversion	R105	Unit Conversion	No
Math	Data Conversion	R106	External Form Conversion	No
Math	Signal Processing	R107	Signal Processing Package	No
Math	Signal Processing	R108	Limiter	No
Math	Signal Processing	R037	Limiter	No
Math	Signal Processing	R038	Limi .r	No
Math	Signal Processing	R160	Limiter	No
Math	Signal Processing	R109	General 1st Order Filter	No
Math	Signal Processing	R161	Tustin Lead-Lag Filter	No
Math	Signal Processing	R110	Second Order Filter	No
Math	Signal Processing	R111	Filter Coefficient	No
Math	Signal Processing	R162	Tustin Lag Filter	No
Math	Signal Processing	R202	Absolute Limiter with Flag	No
Math	Signal Processing	R203	Tustin Integrator with Limit	No
Math	General Purpose	R112	General Math Package	No
Math	General Purpose	R113	Change Calculator	No
Math	General Purpose	R114	Accumulator	No

CATEGORY	SUBCATEGORY	PART #	NAME	Schematic Only
Math	General Purpose	R115	Change Accumulator	No
Math	General Purpose	R116	Interpolation	No
Math	General Purpose	R117	Extrapolation	No
Math	General Purpose	R118	Interpolation Table	No
Math	General Purpose	R119	Interpolation Table	No
Math	General Purpose	R120	Incrementor	No
Math	General Purpose	R121	Decrementor	No
Math	General Purpose	R122	Absolute Value	No
Math	General Purpose	R224	Sign	No
Math	General Purpose	R123	Square Root	No
Math	General Purpose	R124	Integrator	No
Math	General Purpose	R142	Running Average	No
Math	General Purpose	R143	Mean Absolute Difference	No
Math	General Purpose	R144	Mean Value of a Vector	No
Math	Polynomial	R213	Table Lookup	No
Math	Polynomial	R214	Chebyshev	No
Math	Polynomial	R215	Fike	No
Math	Polynomial	R216	Hart	No
Math	Polynomial	R217	Hastings	No
Math	Polynomial	R218	Least Squares	No
Math	Polynomial	R219	Legendre	No
Math	Polynomial	R220	Modified Newton-Raphson	No
Math	Polynomial	R221	Newton-Raphson	No
Math	Polynomial	R222	Taylor Series	No
Math	Polynomial	R223	System Functions	No
Abstract Mechanism	Data Structure	R125	FIFO Buffer	No
Abstract Mechanism	Data Structure	R126	Circular Buffer	No
Abstract Mechanism	Data Structure	R164	FIFO Buffer	No
Abstract Mechanism	Data Structure	R165	Priority Queue	No
Abstract Mechanism	Data Structure	R166	Stack	No
Abstract Mechanism	Data Structure	R167	Unbounded Stack	No
Abstract Mechanism	Process	R127	Finite State Machine Constructor	Yes
Abstract Mechanism	Process	R128	Mealy Machine Constructor	Yes
Abstract Mechanism	Process	R129	Event-Driven Sequencer Constructor	Yes
Abstract Mechanism	Process	R130	Time-Driven Sequencer Constructor	Yes
Abstract Mechanism	Process	R159	Sequence Controller Constructor	Yes
Process Management	Asynchronous Control	R131	Process Controller Constructor	Yes

CATEGORY	SUBCATEGORY	PART #	NAME	Schematic Only
Process Management	Asynchronous Control	R132	Aperiodic Task Shell Constructor	Yes
Process Management	Asynchronous Control	R133	Continuous Task Shell Constructor	Yes
Process Management	Asynchronous Control	R134	Periodic Task Shell Constructor	Yes
Process Management	Asynchronous Control	R135	Data Driven Task Shell Constructor	Yes
Process Management	Communication	R136	Message Checksum	No
Process Management	Communication	R137	Update Exclusion	No
General Utility		R138	Memory Checksum	No
General Utility		R139	Memory Checksum	No
General Utility		R140	Memory Declassification	No
General Utility		R141	Instruction Set Test	No
General Utility		R239	Generic Instantiation Constructor	Yes
Guidance & Control	Autopilot	R048	Integral Plus Proportional (IPP) Gain	No
Guidance & Control	Autopilot	R059	Pitch Autopilot	No
Guidance & Control	Autopilot	R064	Lateral/Directional Autopilot	No
Guidance & Control	Waypoint Steering	R168	Compute Unit Radial Vector	No
Guidance & Control	Waypoint Steering	R169	Compute Segment Unit Normal Vector	No
Guidance & Control	Waypoint Steering	R170	Initialize Steering Vectors	No
Guidance & Control	Waypoint Steering	R171	Update Steering Vectors	No
Guidance & Control	Waypoint Steering	R172	Compute Turn Angle and Direction	No
Guidance & Control	Waypoint Steering	R173	Compute Crosstrack and Heading Error when Turning	No
Guidance & Control	Waypoint Steering	R174	Compute Crosstrack and Heading Error	No
Guidance & Control	Waypoint Steering	R175	Compute Crosstrack and Heading Error when not Turning	No
Guidance & Control	Waypoint Steering	R176	Compute Distance to Current Waypoint	No
Guidance & Control	Waypoint Steering	R177	Comput Turning and Nonturning Distances	No
Guidance & Control	Waypoint Steering	R178	Perform Stop Turn Test	No
Guidance & Control	Waypoint Steering	R179	Perform Start Turn Test	No
Guidance & Control	Waypoint Steering	R180	Perform Start/Stop Turn Test	No
Non-Guidance Control	Air Data	R228	Compute Outside Air Temperature	No
Non-Guidance Control	Air Data	R229	Compute Pressure Ratio	No
Non-Guidance Control	Air Data	R230	Compute Mach	No
Non-Guidance Control	Air Data	R231	Compute Dynamic Pressure	No
Non-Guidance Control	Air Data	R232	Compute Speed of Sound	No
Non-Guidance Control	Air Data	R233	Compute Barometric Altitude	No
Non-Guidance Control	Fuel Con[illegible]	[illegible]	Mach Control	No
		21[illegible]	<== TOTAL NUMBER OF PARTS	

(The reverse of this page is blank)

INITIAL DISTRIBUTION

DTIC-DDAC	2
AUL/LSE	1
FTD/SDNF	1
HQ USAFE/INATW	1
AFWAL/FIES/SURVIAC	1
AFATL/DOIL	2
AFATL/CC	1
AFCSA/SAMI	1
AFATL/CCN	1
AFATL/FXG	10
ASD/RWX	1
SPAWAR (814AB NC #1)	1
SPAWAR (CODE 613)	1
NRL (CODE 5150)	1
ASD/XRX	1
AFWAL/AAA-2	1
HQ AFSC/PLR	3
INST OF DEFENSE ANALYSES	5
STARS JOINT PROGRAM OFFICE	1
USA MATERIEL CMD/AMCDE-SB	1
NAVAL SEA SYS CMD (SEA 61R2)	1
NAVAL AIR DEVELOPMENT CTR (CODE 50C)	1
BMDATC	1
NSWC/CODE N20	1
NAVAL UNDERSEA SYS CTR (CODE 3511)	1
NOSC/CODE 423	1
AFWAL/AAAF-2	1
USA EPG/STEEP-MT-DA	1
AFSC/DLA	1
USA MSL CMD/AMSMI-OAT	1
SPAWAR (CODE 06 NC #1)	1
ASD/EN	1
AD/ENE	1
SD/ALR	1
RADC/COEE	2
NRL/CODE 7590	1
AFSC/SDZD	1
HQ USAF/RDPV	1
AD/ENSM	1
BMO/ENBE	1
ESD/ALS	1
ESD/ALSE	2
AJPO	1
AFATL/AS	1
AFATL/SA	1
AD/XR	1
WIS JPMO/ADT	1
AFWAL/AAAF	1

NASA (CODE RC)	1
NSA	1
NTSC/CODE 251	1
AD/XRB	1
LOCKHEED (DR SURY)	1
HUGHES (MR BARDIN)	1
IBM (MS VESPER)	1
RAYTHEON (WILLMAN)	1
SOFTECH INC (MS BRAUN)	1
MITRE CORP (MR SYLVESTER)	1
AEROSPACE CORP (MR HOGAN)	1
TEXAS INSTRUMENTS (MR FOREMAN)	1
RATIONAL (MR BOOCH)	1
ROCKWELL INTERNATIONAL (MR GRIFFIN)	1
MITRE CORP (MS CLAPP)	1
NOSC (CODE 423)	1
GENERAL DYNAMICS (MR MURRAY)	1
HONEYWELL INC (MS GIDDINGS)	1
HONEYWELL INC (DR FRANKOWSKI)	1
HUGHES DEFENSE SYS DIV (S.Y. WONG)	1
NOSC (MR WASILAUSKI)	1
NAC (N)	1
TASC (MR SERNA)	1
MARTIN MARIETTA (MR CUDDIE)	1
TASC (DR CRAWFORD)	2
SYSCON CORP (DR BRINTZINHOFF)	1
ADA TRAINING SECTION	1
COMPUTER SCI CORP (MR FRITZ)	1
LINKABIT (MR SIMON)	1
RAYTHEON (MR GINGERICH)	1
UNIVERSITY OF COLORADO	1
MCDONNELL AIRCRAFT (MR MCTIGUE)	1
HUGHES AIRCRAFT (MR NOBLE)	1
GENERAL DYNAMICS (MR PRZYBYLINSKI)	1
AJPO (MS CASTOR)	1
NWC (CODE 3922)	1
NAVAIRSYSCOM HQS	1
ASD/ENASF	1
ASD/ENA	5
UNIVERSITY OF TEXAS	1
MARTIN MARIETTA (MR SORONDO)	1
MCDONNELL DOUGLAS (DR MCNICHOLL)	2
GENERAL DYNAMICS (MR SCHNELKER)	1
TRW (MR SHUGERMAN)	1
RATIONAL (MR HAKE)	1
WESTINGHOUSE (MR GREGORY)	2
GENERAL DYNAMICS (DR TABER)	1
BOEING COMMERCIAL AIRPLANE CO	1

INITIAL DISTRIBUTION (CONTINUED)

ROCKWELL INTERNATIONAL (MIKULSKI)	1
BOEING AEROSPACE (MR HADLEY)	1
IBM (MR MCCAIN)	1
FSU (DR BAKER)	1
AEROSPACE CORP (MR LUBOFSKY)	1
DATA GENERAL (MR DAMASHEK)	1
SDC (MR HERMANN)	1
WINTEC (MR CONNEL)	1
NORTHROP (MR OHLSEN)	1
ARC (MR ROBERSON)	1
GRC (DR ALBRITTON)	1
AFWL/NTSAC	1
USA MSL CMD/SCI INFO CTR	1
NSWC/TECH LIB	1
NWC (CODE 343)	1
OO-NAVAL RESEARCH (CODE 784DL)	1
NAVAL POSTGRAD SCHOOL (CODE 1424)	1
DEFENSE COMMUNICATIONS AGENCY	1
NASA AMES RESEARCH CTR (CHAPMAN)	1
NASA LANGLEY RESEARCH CTR (MOTLEY)	2
RAND CORP/AFELM	1
AVCO SYS DIV/RESEARCH LIB	1
AVCO-EVERETT RES LAB/TECH LIB	1
FAIRCHILD IND/INFO CTR	1
GENERAL DYNAMICS, CONVAIR DIV	1
GENERAL DYNAMICS, FT WORTH DIV	1
HONEYWELL/TECH LIB	1
HUGHES AIRCRAFT/TECH LIB	1
HUGHES AIRCRAFT/MSL SYS GP/TECH LIB	1
LOCKHEED/TECH LIB	1
LOCKHEED/TECH INFO CTR	1
MARTIN MARIETTA/TECH LIB	1
MCDONNELL DOUGLAS/TECH LIB	1
MCDONNELL DOUGLAS/LIB SVCS	1
NORTHROP CORP/AIRCRAFT DIV	1
RAYTHEON/TECH LIB	1
VOUGHT CORP/LIB	1
SDC (MR MILLAR)	1
INTERMETRICS (MR ZIMMERMANN)	1
ROCKWELL/TECH INFO	1
TRW (MR MUNGLE)	1
AT&T (MR MAY)	1
AVCO SYS TEXTRON (MR SOHN)	1
BOEING ELECTRONICS (MR MEDAN)	1
USA CECOM/DRSEL-TCS-MCF	1
USA CECOM/DRSEL-TCS-ADA	1
UNIVERSITY OF WASHINGTON	1
AFATL/GRC	1
USA-ARDC	1
ROCKWELL (MS KIM)	1
SCIENCE APPLICATIONS INTERNATIONAL (MR STUTZKE)	1
NSWC (U-33)	1
COMPUTER SOFTWARE & SYSTEMS	1
GRUMMAN DATA SYSTEMS (MR MARKMAN)	1
AFWAL/FIGL	1
AFWAL/FIGX	1
AFWAL/AARI-1	1
EMERSON ELECTRIC (MR BYRNES)	1
TASC (MR JAZMINSKI)	1
E-SYSTEMS (MR SNODGRASS)	1
NTSC (CODE 742)	1
UNIVERSITY OF ALABAMA	1
MCDONNELL DOUGLAS (MR VILLACHICA)	1
SANDERS ASSOCIATES (MR FRY)	1
AFSC/PLR	1
AFSC/DLA	1
WESTINGHOUSE (MR SQUIRE)	1
NAVAIR SYSCOM (AIR 54662)	1
BOEING AEROSPACE (MR BOWEN)	1
GOULD INC (DR ARORA)	1
LOCKHEED (MR PINCUS)	1
INTERMETRICS (MR BROIDO)	1
GENERAL ELECTRIC (MS MICKEL)	1
HUGHES AIRCRAFT (DR HUANG)	1
SYSTEMS DEVELOPMENT (MR SIMOS)	1
GENERAL DYNAMICS (MR WARNER)	1
CARNEGIE-MELLON UNIVERSITY	1
COMPUTER TECH ASSOCIATES (HEYLIGER)	1
NORTHROP CORP (MR SWAN)	1
CNI SOFTWARE (SINGER KEARFOTT DIV)	1
LOCKHEED (MR COHEN)	1
MCDONNELL DOUGLAS (SANDY COHEN)	2
DRAPER LABORATORY (MR DAVID)	1
DRAPER LABORATORY (DR DEWOLF)	1
GOULD INC (MR WILLIS)	1
ADVANCED TECHNOLOGY (MR COOPER)	1
HQ USAF/RDST	1
AFPRO/EN	1
AFPRO/TRW/EPP	1
LOCKHEED (MR DORFMAN)	1
HONEYWELL (MR LANE)	1
SPERRY A&MG (MR ROSS)	1
UNITED TECHNOLOGIES (STOLZENTHALER)	1
LTV AEROSPACE & DEFENSE	1
EL66 B 4610 (MR FREEMAN)	1
SYSTEMS DEVELOPMENT CORP (ATCHLEY)	1

INITIAL DISTRIBUTION (CONCLUDED)

ROCKWELL (DILLHUNT)	1
IBM FED SYSTEMS DIV (MR ANGIER)	1
JET PROPULSION LAB (MR KRASNER)	1
MARTIN MARIETTA ORLANDO AEROSPACE	1
GENERAL ELECTRIC (MR DELLEN)	1
AFWAL/POFIC	1
LOCKHEED (READY)	1
BOEING AEROSPACE (MR DOE)	1
FORD AEROSPACE (DR FOX)	1
LOCKHEED SOFTWARE TECH CTR (LYONS)	1
IBM (MR BENESCH)	1
MITRE CORP (MR SHAPIRO)	1
MITRE CORP (MR MAGINNIS)	1
GRUMMAN AEROSPACE (MR POLOFSKY)	1
MCDONNELL DOUGLAS (KAREN L SAYLE)	1
MCDONNELL DOUGLAS (GLENN P LOVE)	1
ROME AIR DEVELOPMENT CTR (CHRUSCICKI)	1
RAYTHEON EQUIPMENT DIV (MS SILLERS)	1
AT&T BELL LABS (MS STETTER)	1

(The reverse of this page is blank)

THIS REPORT HAS BEEN DELIMITED AND CLEARED FOR PUBLIC RELEASE UNDER DOD DIRECTIVE 5200.20 AND NO RESTRICTIONS ARE IMPOSED UPON ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

B102657

ERRATA

AFATL-TR-85-93

VOLUMES I, II, and III

COMMON ADA® MISSILE PACKAGES (CAMP)

FINAL REPORT

AIR FORCE ARMAMENT LABORATORY

EGLIN AIR FORCE BASE, FLORIDA

32542-5434

1. DD Form 1473:

   In Block 16, add the following:

   "Methodology used in this report does not constitute computer software as defined in AFR 300-6."

2. This errata is unclassified.

Lynn S. Marrs

LYNN S. MARRS
Chief, Technical Reports Section

SUPPLEMENTARY

INFORMATION

DEPARTMENT OF THE AIR FORCE
WRIGHT LABORATORY (AFSC)
EGLIN AIR FORCE BASE, FLORIDA, 32542-5434

ERRATA
AD-B102656

REPLY TO ATTN OF: MNOI

13 Feb 92

SUBJECT: Removal of Distribution Statement and Export-Control Warning Notices

TO: Defense Technical Information Center
ATTN: DTIC/HAR (Mr William Bush)
Bldg 5, Cameron Station
Alexandria, VA 22304-6145

1. The following technical reports have been approved for public release by the local Public Affairs Office (copy attached).

Technical Report Number	AD Number
1. 88-18-Vol-4	ADB 120 251
2. 88-18-Vol-5	ADB 120 252
3. 88-18-Vol-6	ADB 120 253
4. 88-25-Vol-1	ADB 120 309
5. 88-25-Vol-2	ADB 120 310
6. 88-62-Vol-1	ADB 129 568
7. 88-62-Vol-2	ADB 129 569
8. 88-62-Vol-3	ADB 129-570
9. 85-93-Vol-1	ADB 102-654
10. 85-93-Vol-2	ADB 102-655
11. 85-93-Vol-3	ADB 102-656
12. 88-18-Vol-1	ADB 120 248
13. 88-18-Vol-2	ADB 120 249
14. 88-18-Vol-7	ADB 120 254
15. 88-18-Vol-8	ADB 120 255
16. 88-18-Vol-9	ADB 120 256
17. 88-18-Vol-10	ADB 120 257 *
18. 88-18-Vol-11	ADB 120 258
19. 88-18-Vol-12	ADB 120 259

2. If you have any questions regarding this request call me at DSN 872-4620.

Lynn S. Wargo
LYNN S. WARGO
Chief, Scientific and Technical
Information Branch

1 Atch
AFDTC/PA Ltr, dtd 30 Jan 92

ERRATA

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS AIR FORCE DEVELOPMENT TEST CENTER (AFSC)
EGLIN AIR FORCE BASE, FLORIDA 32542-5000

REPLY TO ATTN OF: PA (Jim Swinson, 882-3931)

30 January 1992

SUBJECT: Clearance for Public Release

TO: WL/MNA

The following technical reports have been reviewed and are approved for public release: AFATL-TR-88-18 (Volumes 1 & 2), AFATL-TR-88-18 (Volumes 4 thru 12), AFATL-TR-88-25 (Volumes 1 & 2), AFATL-TR-88-62 (Volumes 1 thru 3) and AFATL-TR-85-93 (Volumes 1 thru 3).

Virginia N. Pribyla

VIRGINIA N. PRIBYLA, Lt Col, USAF
Chief of Public Affairs

AFDTC/PA 92-039